Machine Learning for Decision Making

Amir Sani

UNIVERSITY OF LILLE 1
Ecole Doctorale Sciences Pour l'Ingénieur
Laboratoire Paul Painlevé
CNRS U.M.R. 8524 , 59655 Villeneuve d'Ascq Cedex
INRIA Lille – Nord Europe

THÈSE de Doctorat

Specialty: Mathématiques Appliquées

l'Apprentissage Automatique pour la prise de Décisions

presented by
Amir SANI

Defended in Villeneuve d'Ascq on May 12th, 2015 in front of a jury composed of:

Rapporteur :	Jean-Yves Audibert	-	Capital Fund Management
Rapporteur :	Mark Herbster	-	University College London
Directeur :	Rémi Munos	-	Google DeepMind, INRIA
Co-Directeur :	Alessandro Lazaric	-	INRIA
Examinateur :	Cristophe Biernacki	-	University Lille 1
Examinateur :	Balázs Kégl	-	Laboratoire de l'Accélérateur Linéaire, CNRS
Examinateur :	Antoine Mandel	-	Paris School of Economics, CNRS
Examinateur :	Gilles Stoltz	-	HEC Paris, CNRS

Résumé court en français:

La prise de décision stratégique concernant des ressources de valeur devrait tenir compte du degré d'aversion au risque. D'ailleurs, de nombreux domaines d'application mettent le risque au cœur de la prise de décision. Toutefois, ce n'est pas le cas de l'apprentissage automatique. Ainsi, il semble essentiel de devoir fournir des indicateurs et des algorithmes dotant l'apprentissage automatique de la possibilité de prendre en considération le risque dans la prise de décision. En particulier, nous souhaiterions pouvoir estimer ce dernier sur de courtes séquences dépendantes générées à partir de la classe la plus générale possible de processus stochastiques en utilisant des outils théoriques d'inférence statistique et d'aversion au risque dans la prise de décision séquentielle. Cette thèse étudie ces deux problèmes en fournissant des méthodes algorithmiques prenant en considération le risque dans le cadre de la prise de décision en apprentissage automatique. Un algorithme avec des performances de pointe est proposé pour une estimation précise des statistiques de risque avec la classe la plus générale de processus ergodiques et stochastiques. De plus, la notion d'aversion au risque est introduite dans la prise de décision séquentielle (apprentissage en ligne) à la fois dans les jeux de bandits stochastiques et dans l'apprentissage séquentiel antagoniste.

English Title: Machine Learning for Decision-Making Under Uncertainty

Short English Abstract:

Strategic decision-making over valuable resources should consider risk-averse objectives. Many practical areas of application consider risk as central to decision-making. However, machine learning does not. As a result, research should provide insights and algorithms that endow machine learning with the ability to consider decision-theoretic risk. In particular, in estimating decision-theoretic risk on short dependent sequences generated from the most general possible class of processes for statistical inference and through decision-theoretic risk objectives in sequential decision-making. This thesis studies these two problems to provide principled algorithmic methods for considering decision-theoretic risk in machine learning. An algorithm with state-of-the-art performance is introduced for accurate estimation of risk statistics on the most general class of stationary–ergodic processes

and risk-averse objectives are introduced in sequential decision-making (online learning) in both the stochastic multi-arm bandit setting and the adversarial full-information setting.

Mots clés: Apprentissage Automatique, Algorithme d'apprentissage incrémental, Prise de Décision (statistique), Bootstrap (statistique), Risque, Prise de Décision, Optimisation, Bandit manchot (Mathématiques)

English Keywords: Machine Learning, Online Learning, Sequential Decision-Making, Bootstrap, Risk-Aversion, Decision-Making, Multi-Arm Bandit, Learning with Expert Advice

For Lorène. To our Future

Acknowledgments

Thank you to all the people that have made a significant impact on my choices simply as a matter of their generous character. In particular, I appreciate the sacrifices made by my parents to give me such a wonderful perspective on life. I look forward to working with Antoine Mandel at the Paris School of Economics to bring machine learning and online machine learning to macroeconomics.

I want to thank many people for their collaboration and contributions along this journey. Rémi Munos for his support and supervision. Alessandro Lazaric for taking a chance on a student from industry. I appreciated your patience, perseverance and focus in helping me realize my goals. Daniil Ryabko for our insightful conversations that provided so much intuition, and for introducing me to the measure R. Gergely Neu for understanding my ramblings on experts and benchmarks. Your welcoming nature made it easy to discover new ideas. Ronald Ortner for spending his sabbatical at SequeL. I appreciated our discussions and your generosity. Nathan Korda for our frequent discussions and helpful comments on this thesis. Alexandra Carpentier for her generous support and review of this thesis. Lorène Sani and Ariana Sani for re-reading my thesis for grammar and stylistic mistakes. Bilal Piot for reviewing my French abstract and keywords. Mark Herbster and Jean-Yves Audibert for acting as rapporteurs for this thesis and taking the time to provide helpful feedback and questions. In particular, I was very impressed by the quality of reviews and did not expect such insightful comments on three seemingly distant areas which were brought together under a single theme. Thank you for Cristophe Biernacki, Balázs Kégl, Antoine Mandel and Gilles Stoltz for being part of my jury. Thank you to the SequeL team for their support and INRIA for providing a great setting for the PhD. Finally, thank you to the Grid 5000 network in France for generously providing the compute resources necessary to produce so many interesting contributions.

A special thank you to my wife, Lorène Sani. You inspire me with your strength, character, intelligence and vision.

Abstract

Strategic decision-making over valuable resources should consider risk-averse objectives. Many practical areas of application consider risk as central to decision-making. However, machine learning does not. As a result, research should provide insights and algorithms that endow machine learning with the ability to consider decision-theoretic risk. The thesis highlights the impact of risk-averse objectives in machine learning, while the algorithms are meant to introduce principled methods for integrating decision-theoretic risk through accurate estimation of risk statistics and risk-averse objectives in sequential decision-making (online learning). Many machine learning algorithms for decision making focus on estimating performance with regard to the expectation. In many practical problems, measuring performance according to the expectation may not be very meaningful. This thesis provides principled algorithmic methods for accurate estimation of risk statistics on the most general class of stationary–ergodic processes and risk-averse objectives are introduced in sequential decision-making (online learning) in both the stochastic multi-arm bandit setting and the adversarial full-information setting.

Author's Relevant Publications

A. Sani, A. Lazaric, D. Ryabko, *The Replacement Bootstrap for Dependent Data,* 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, China, (June 2015).

A. Sani, G. Neu, A. Lazaric, *Exploiting Easy Data in Online Optimization,* Neural Information Processing Systems (NIPS), Montreal, Canada, (December 2014).

A. Sani, A. Lazaric, D. Ryabko, *Information Theoretic Bootstrapping for Dependent Time Series,* Neural Information Processing Systems (NIPS) Workshop in Modern Nonparametric Methods in Machine Learning, Lake Tahoe, Nevada, (December 2013).

A. Sani, A. Lazaric, R. Munos, *Risk Aversion in Multi-Arm Bandits,* Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, (December 2012).

A. Sani, A. Lazaric, R. Munos, *Risk Averse Multi-Arm Bandits,* Markets, Mechanisms and Multi-Agent Models, International Conference in Machine Learning (ICML) Workshop, Edinburgh, Scotland, (June 2012).

List of Figures

1.1 Fully specified distributions. 3

2.1 The Bootstrap Protocol 14
2.2 i.i.d. Bootstrap Algorithm 15
2.3 Block Bootstrap Algorithm 16
2.4 Markov Bootstrap Algorithm 18
2.5 A simple 1-Markov chain. 21
2.6 Pseudo-code of the $\mathcal{R}$-Boot algorithm. 26
2.7 Computation of other replacement points (i.e., $l \geq 1$). 32
2.8 (LEFT) Sequences generated from the true process P along with an illustration of the maximum drawdown (RED) on the black sequence. (RIGHT) Bootstrap sequences generated by $\mathcal{R}$-Boot from the black trajectory. 39
2.9 MSE on the FBM process for the maximum drawdown statistic. 42
2.10 Bootstrap estimation performance for the mean (LEFT) and standard deviation(RIGHT). 43
2.11 Sensitivity analysis of circular block Bootstrap, Markov Bootstrap, and $\mathcal{R}$-Boot with respect to their parameters (block width, Markov order, and number of replacements R as a percentage of T) in the FBM experiment for $T = 200$ (notice the difference in scale for circular block Bootstrap). 44
2.12 Maximum drawdown MSE performance (with standard errors) on multiple real datasets. 45
2.13 Sensitivity to parameters for different Bootstrap methods for the USDCHF currency (notice the difference in scale for circular block Bootstrap). 46

3.1 *UCB1* 55
3.2 *UCB-V* 56

3.3 Pseudo-code of the *MV-LCB* algorithm. 65
3.4 Pseudo-code of the *ExpExp* algorithm. 73
3.5 *MV-LCB* Regret (LEFT) and worst-case performance of *MV-LCB* versus *ExpExp*, for different values of $T \times 10^3$(RIGHT). 77
3.6 Regret $\mathcal{R}_T$ of *MV-LCB*. 78
3.7 Regret $\mathcal{R}_T$ of *ExpExp*. 79
3.8 Configuration 1 and Configuration 2. 80
3.9 Risk tolerance sensitivity of *MV-LCB* and *ExpExp* for *Configuration 1.* . . 81
3.10 Risk tolerance sensitivity of *MV-LCB* and *ExpExp* for *Configuration 2.* . . 81

4.1 Online Learning Protocol 91
4.2 Follow the Leader (*FTL*) 92
4.3 HEDGE 94
4.4 PROD 96
4.5 Mean–Deviation [Even-Dar et al., 2006] 100
4.6 Variance–Loss [Warmuth and Kuzmin, 2012] 101
4.7 *D*-PROD [Even-Dar et al., 2008] 104
4.8 $(\mathcal{A}, \mathcal{B})$-PROD 108
4.9 $(\mathcal{A}, \mathcal{B})$-PROD (Anytime) 110
4.10 Impact of PROD update. 113
4.11 Performance comparison of *FTL* and *HEDGE* on easy versus hard data. 115
4.12 Performance comparison of $(\mathcal{A}, \mathcal{B})$-PROD, *FTL* and *HEDGE* on easy versus hard data. 117
4.13 Hand tuned loss sequences from de Rooij et al. [2014] 125

Contents

1 Introduction **1**

2 The Replacement Bootstrap **9**

1 Introduction . . . 9
2 Preliminaries . . . 13
3 The Bootstrap . . . 14
3.1 Block Bootstrap . . . 16
3.2 Markov Bootstrap . . . 18
4 The Replacement Bootstrap . . . 19
4.1 Comparing the Replacement Probability to the Prediction Probability . . . 21
4.2 The $\mathcal{R}$-Boot Algorithm . . . 23
4.3 Impact of Replacement Parameter R . . . 27
4.4 Implementation details . . . 27
4.5 A Comparison with Markov Bootstrap . . . 34
4.6 Theoretical Guarantees . . . 35
5 Empirical Evaluation . . . 39
5.1 Currency Datasets . . . 45
6 Conclusions . . . 46
7 Future Work . . . 47

3 Risk Averse Multi-Arm Bandits **49**

1 Introduction . . . 49
2 The Multi–Arm Bandit Problem . . . 53
2.1 Notation, Setting and Definitions . . . 53
2.2 Optimism in the Face of Uncertainty Principle . . . 54
3 Mean–Variance Multi–arm Bandit . . . 57
3.1 Additional Notation, Setting and Definitions . . . 57
4 Mean–Variance Lower Confidence Bound Algorithm . . . 65

4.1 Theoretical Analysis . . . 67
4.2 Worst–Case Analysis . . . 71
5 Exploration–Exploitation Algorithm . . . 72
5.1 Theoretical Analysis . . . 73
6 Numerical Simulations . . . 77
7 Sensitivity Analysis . . . 78
8 Discussion . . . 82
9 Conclusions . . . 83
10 Subsequent Work . . . 85

4 Online Learning with a Benchmark 87

1 Introduction . . . 87
2 Preliminaries . . . 91
2.1 Online Learning with Full Information . . . 91
2.2 Prediction with Expert Advice . . . 93
2.3 Weighted Majority Algorithms . . . 93
3 Risk in Online Learning . . . 99
3.1 Risk Sensitive Online Learning . . . 99
3.2 Online Variance–Loss Minimization . . . 102
3.3 Risk to the *Best* versus Risk to the *Average* . . . 103
4 Online Learning with a *flexible* Benchmark . . . 105
4.1 $(\mathcal{A}, \mathcal{B})$-Prod . . . 106
4.2 Discussion . . . 111
5 Applications . . . 113
5.1 Prediction with expert advice . . . 114
5.2 Tracking the best expert . . . 117
5.3 Online convex optimization . . . 119
5.4 Learning with two-points-bandit feedback . . . 121
6 Empirical Results . . . 123
6.1 Settings . . . 124
7 Conclusions . . . 126

5 Conclusion 127

Bibliography 129

Chapter 1

Introduction

Risk is central to decision-making in many domains. A non-exhaustive list includes economics [Knight, 2012], insurance [Dorfman and Cather, 2012], banking [Bessis, 2011], portfolio management [Grinold and Kahn, 1999], investments [Crouhy et al., 2014], financial institution risk [Hull, 2012], enterprise risk [Lam, 2014], operations management [Ritchie and Angelis, 2011], business management [Pritchard et al., 2014], engineering [Ayyub, 2014] and environmental science [O'Riordan, 2014]. Machine learning applications to both decision-making and decision-support are growing. Further, with each successful application, learning algorithms are gaining increased autonomy and control over decision-making. As a result, research into intelligent decision-making algorithms continues to improve. For example, the Stanford Research Institute's Cognitive Assistant that Learns and Organizes project focuses on creating an intelligent desktop assistant with the capability to learn and reason. The aim is for an intelligent virtual assistant to autonomously handle tasks. Another example is Watson, which after outperforming the top players in the human question–answer game Jeopardy, was repositioned as an intelligent decision support tool. Current application areas include financial planning, drug research, medicine and law. Many of these application domains deal with an underlying randomness of choice distributions that is unknown *a priori*. Specific example problems include fundamental infrastructure repairs [Li et al., 2014], predicting severe weather [McGovern et al., 2014], predicting aviation turbulence [Williams, 2014], tax audits [Kong and Saar-Tsechansky, 2014] and privacy breach detection [Menon et al., 2014]. The performance of machine learning algorithms directly depends on how explicit the unique aspects of the domain are formalized [Rudin and Wagstaff, 2014]. Considering the increasing autonomy of machine learning algorithms in decision-making, it is natural to consider notions

of decision-theoretic risk with respect to this unknown randomness. When applied to decision-making, machine learning algorithms do not generally consider risk objectives. Including risk formally within the learning objective allows the algorithm to weight decisions according to their risk. This thesis introduces machine learning algorithms that consider such risk-averse objectives. In particular, this thesis considers accurate estimation of complex risk statistics on dependent processes, managing risk-aversion under partial-information in sequential decision-making, and exploiting full-information sequential decision-making with the protection of a benchmark.

This thesis studies decision-theoretic risks and does not directly study, propose or evaluate risk measures (for a full review of statistical measures of risk, please see Schied [2006], Rockafellar [2007]). The aim is to highlight risk-averse objectives in machine learning, when machine learning is used for decision-making or decision-support. The concept of risk covers many domains with diverse interpretations (for a full review of decision-theoretic risk, please see e.g., Peterson [2009], Gilboa [2009]). Willett [1901] referred to it as an "objectified uncertainty regarding the occurrence of an undesirable event". Knight [1921] described risk as "knowing with certainty the mathematical probabilities of possible outcomes", and uncertainty as when "the likelihood of outcomes cannot be expressed with any mathematical precision". Machine learning literature often refers to the "risk" of learning, but this is related to the sub-optimal performance due to the uncertainty intrinsically present in (random) samples [Hastie et al., 2009]. This thesis only considers risk observing objectives and their impact on machine learning algorithms used for decision-making. Two popular, well-studied, risk objectives come from economics and finance. The primary economic model for decision-making is expected utility theory [Neumann and Morgenstern, 1947]. Expected utility theory maps utilities over observations and requires an explicit utility function. Extensions introduce alternatives for how utility functions are specified. The general principle of mapping values to utilities remains. The financial literature bases its decision-making model on a simpler principle of trading-off risk versus reward. Initially introduced in the literature by Markowitz [1952] as the Mean–Variance model, the risk–reward

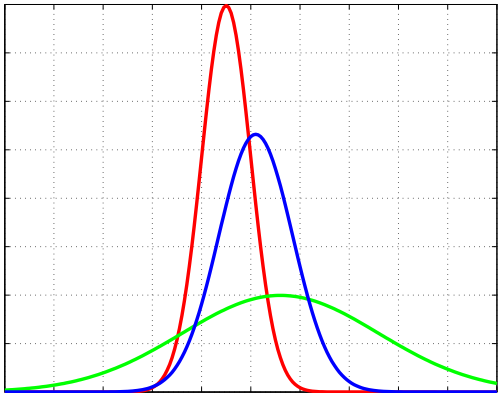

Figure 1.1: Fully specified distributions.

principle is a natural way to characterize a preference over risks. Extensions to the Mean–Variance model maintain the risk–reward trade-off, while considering alternatives to the mean and variance. The advantage in using the Mean–Variance model is that it is composed of unbiased estimators and allows for a natural and intuitive notion of a risk preference [Markowitz, 2014]. It also allows a consistent preference ordering over the set of all probability distributions with bounded support in a finite real interval [Chiu, 2007]. Markowitz [2014] specifically showed consistency for the Mean–Variance objective for quadratic utilities with arbitrarily distributed rewards, arbitrary utilities with normally distributed rewards and log normally distributed rewards according to a Pratt [1964] coefficient of absolute risk aversion. Consider the evaluation of the fully specified distributions in Figure 1.1. The standard expectation maximization objective used in machine learning algorithms for decision-making prefers the **Green** distribution. The variance-averse objective prefers the **Red** distribution, and the Mean–Variance objective prefers the **Blue** distribution.

These models can be linked by assuming that the specific Mean–Variance model approximates expected utility [Levy and Markowitz, 1979]. Though both models are often criticized, subsequent extensions have not replaced their underlying principles. Additionally, asset management and trading models within finance also consider "hedging" rewards through measurable benchmarks [Bychuk and Haughey, 2011]. The principle is to reduce risk by anticipating the possible underperformance to some acceptable benchmark performance. Rather than

mapping values to utilities or trading-off rewards against risk, "hedging" focuses on reducing risk *exposure*. Under this model, risk is limited to the performance of the benchmark. Statistical tools are required to reduce uncertainty and improve decision-making.

Consider the case where only a limited number of dependent samples are available. Estimation of risk-averse objectives on limited samples from a dependent sequence is challenging, yet critical for decision-making under uncertainty. Many estimators are designed for specific statistics or make restrictive structural assumptions. They also require accurate parameter selection to guarantee consistent estimates. This can be challenging when the true measure of the statistic is unknown or the full correlation structure of the data is unspecified. Further, without a correct model, these procedures may fail. In some cases, simple asymptotic estimators, that only rely on the samples, provide the most efficient estimates. In particular, independent and identically distributed (i.i.d.) data has no correlation structure, so unbiased estimates for simple moment statistics, such as the mean and variance, are fast. Dependent processes can be much more challenging due to their correlation structure. This additional structure might result in complex behaviors that might not even be revealed, especially in short samples. Risk statistics, such as the "max" or "min" of a distribution, can also increase estimation difficulty. These challenges might require much longer observation sequences for accurate estimation, or multiple independent samples of dependent observation sequences, which may not be possible. One example statistic is the maximum drawdown, which is much harder to estimate because it is an extremum statistic for a distribution conditioned on the ordering and length of a sequence. It measures the distance between a peak and subsequent nadir over an observation sequence (for more information, please see e.g., Casati and Tabachnik [2013]). Further, restrictive assumptions on the process limit applicability of estimation tools in the case where the characteristics of the process are unknown and limit consistency to specific processes. Careful selection of estimation tools is required when decision-making to avoid restricting the measurability of statistics or events of interest. Chapter 2 presents a novel nonparametric Bootstrap approach based

on replacements and an information-theoretic iterative Bootstrap algorithm that applies to the most general class of dependent processes possible, with performance validated on the challenging maximum drawdown statistic.

As noted earlier, sequential decision-making algorithms rely on policies to evaluate choices. We study the impact of risk-averse objectives on sequential decision-making by studying two information regimes. First, Chapter 3 studies how policies manage risk-averse objectives in the partial information setting, where observations are only revealed from selected choices. This setting intrinsically captures the exploration–exploitation dilemma, which is the challenge of exploring choices to improve estimation confidence, while exploiting the best choice observed so far. A natural choice for this study is the stochastic environment, which fixes the distributions generating observations and does not give the environment enough power to confound our results. Next, Chapter 4 studies the full-information setting. It is common to use parametric models or make restrictive assumptions on the process generating observations in this setting, so we study an adversarial environment, where no statistical assumptions are made on the process. This allows us to consider any possible class of processes generating observations. Existing algorithms applicable to adversarial full-information setting can become quite complicated and *ad hoc* depending on the particular problem setting. These special-purpose policies may or may not consider an intuitive notion of decision-theoretic risk and may ultimately be limited by their particular application. Chapter 4 introduces an intuitive and flexible structure that lower bounds risk to a fixed, changing or adaptive benchmark, that can even learn, for any possible process, without restrictions on its problem setting.

Three specific problems are studied in this thesis, with state-of-the-art algorithms presented in each case. First, accurate estimation of complex statistics for stationary processes, where an algorithm is introduced that significantly outperforms all state-of-the-art nonparametric approaches on a complex dependent process and several real datasets. Next, the problem of managing a risk-averse objective while managing the exploration–exploitation dilemma, where two algorithms are presented. Finally, the problem of risk-aversion in the most general

adversarial full-information setting, where a flexible state-of-the-art algorithm is introduced to provide a principled and flexible structure to "hedge" risk with a benchmark.

Structure of the Thesis

Chapter 2: The Replacement Bootstrap

Applications that deal with time-series data often require evaluating complex statistics for which each time series is essentially one data point. When only a few time series are available, Bootstrap methods are used to generate additional samples that can be used to evaluate empirically the statistic of interest. In this chapter, we introduce a novel replacement Bootstrap principle and $\mathcal{R}$-Boot, an iterative replacement Bootstrap algorithm, which is shown to have some asymptotic consistency guarantees under the only assumption that the time series are stationary and ergodic. This contrasts previously available results that impose mixing or finite-memory assumptions on the data. $\mathcal{R}$-Boot is empirically evaluated on both simulated and real datasets, demonstrating its capability on a practically relevant and complex extrema statistic.

Chapter 3: Risk-Averse Multi-Arm Bandits

Stochastic multi–armed bandits solve the Exploration–Exploitation dilemma and ultimately maximize the expected reward. Nonetheless, in many practical problems, maximizing the expected reward is not the most desirable objective. In this chapter, we introduce a novel setting based on the principle of risk–aversion where the objective is to compete against the arm with the best risk–return trade–off. This setting proves to be intrinsically more difficult than the standard multi-arm bandit setting due in part to an exploration risk which introduces a regret associated to the variability of an algorithm. Using variance as a measure of risk, we introduce two new algorithms, investigate their theoretical guarantees, and report preliminary empirical results. While *MV-LCB* shows a small regret of order

$\mathcal{O}\left(\frac{\log T}{T}\right)$ on "easy" problems, we showed that it has a constant worst–case regret. On the other hand, we proved that *ExpExp* has a vanishing worst–case regret at the cost of worse performance on "easy" problems. To the best of our knowledge this is the first work introducing risk–aversion in the multi–armed bandit setting and it opens a series of interesting questions.

Chapter 4: Online Learning with a Benchmark

We consider the problem of online optimization, where a learner chooses a decision from a given decision set and suffers some loss associated with the decision and the state of the environment. The learner's objective is to minimize its cumulative regret against the best fixed decision *in hindsight*. Over the past few decades numerous variants have been considered, with many algorithms designed to achieve sub-linear regret in the worst case. However, this level of robustness comes at a cost. Proposed algorithms are often over-conservative, failing to adapt to the *actual* complexity of the loss sequence which is often far from the worst case. In this chapter we introduce $(\mathcal{A}, \mathcal{B})$-PROD, a general-purpose algorithm which receives a learning algorithm $\mathcal{A}$ and a benchmark strategy $\mathcal{B}$ as inputs and guarantees the best regret between the two. We derive general theoretical bounds on the regret of the proposed algorithm. Further, we evaluate the algorithm with the benchmark set to algorithms that exploit "easy" data, with worst-case protection provided by the structure of $(\mathcal{A}, \mathcal{B})$-PROD. Then we discuss its implementation in a wide range of applications, notably solving the COLT open problem of learning with shifting experts. Finally, we provide numerical simulations in the setting of prediction with expert advice with comparisons to the state-of-the-art.

Chapter 2

The Replacement Bootstrap

Contents

1 Introduction 9

2 Preliminaries 13

3 The Bootstrap 14

4 The Replacement Bootstrap 19

5 Empirical Evaluation 39

6 Conclusions 46

7 Future Work 47

1 Introduction

Decision making requires full distributional knowledge for each of the actionable choices *a priori* [Peterson, 2009] to execute a policy. In practice, it is not always the case that full distributional knowledge is available. Policy evaluation can be problematic when statistical properties of the process are unknown, and must be estimated. In this chapter, we remove decision-making and study estimating risk over a general class of processes and complex statistics for decision support.

In many practical applications, such as in finance [Chatfield, 2013], there is generally only a single sequence available for analysis, which is only one of many possible histories that *could have been generated* by the underlying process. Sometimes it is also the case that these sequences are composed of multiple regimes or complex behaviors that are not i.i.d. or stationary, but the result of linear and

nonlinear dependencies. Specific patterns or behaviors in the resulting *dependent* sequence can be very hard to analyze. In some cases, the sequence length is too short to fully exhibit these complex dynamics or these patterns are low probability events that require very long histories to appear. In many practical applications, such as measuring rare events in financial time series [Taleb, 2007] or rare sequence variants in genome sequencing [Steinthorsdottir et al., 2014, Hu et al., 2014], this is not feasible. As a result, when only a single short dependent sequence is observed, estimating complex statistics can be very challenging.

In absence of the process or additional samples, the Bootstrap principle treats the sample distribution as the true distribution, approximating the variability of the true distribution by sampling, with replacement, from the observed sample. The Bootstrap principle achieves excellent estimation performance without making restrictive limiting assumptions on the process. The original i.i.d. Bootstrap [Efron, 1979] assumes independence of sample observations, so it is inconsistent on dependent data [Lahiri, 1999]. Dependent data variations are a popular approach to estimating complex statistics from a single short dependent sequence of observations. They apply more generally, while only requiring strong exponential mixing rates, where the correlations in observation data decay at exponential rates with a growing sample size [Lahiri, 2003] (for a comprehensive review of these methods, see e.g., Berkowitz and Kilian [2000], Bose and Politis [1992], Bühlmann [2002], Lahiri [2003], Härdle et al. [2003], Hongyi Li and Maddala [1996], Politis [2003], Kreiss and Lahiri [2012], Paparoditis and Politis [2009], Ruiz and Pascual [2002]). Two popular nonparametric approaches for dependent data are the *block* and Markov Bootstrap. Block methods directly generalize the Bootstrap principle to dependent data, where the original sequence is segmented into blocks and these blocks are randomly sampled with replacement to construct each bootstrap sequence. The ability of these methods to model serial dependence depends on accurate block width selection (notice that setting the block width to 1 reduces these methods to the original i.i.d. Bootstrap, which does not model serial dependence). The Markov Bootstrap assumes that a Markov model generates the sequence [Kulperger and Rao, 1989] and exceeds the optimal performance of the

block Bootstrap [Horowitz, 2003] only when the process generating the sequence is Markov and the correct model size is specified.

In the case of short sequences, performance for both of these methods depend on the dependency structure of the process decaying inside the realized observation sequence. Correlations between realizations must go to zero inside the realized observation sequence for these methods to work. When this is not the case, and the dependency extends beyond the sequence length, both methods fail. Complex observation sequences are common in practice, where the dependency between data points extends beyond the realized sequence. In this case, standard tools are limited. It might be the case that the risk measure only appears infrequently or that the statistic is a complex extremum statistic. This can be problematic when there are not enough realizations to fully characterize the statistical properties of the risk measure. This is akin to having little to no samples to measure from. When a policy must evaluate choices according to such a statistic, this limitation can be quite problematic. As a result, we choose to drop this restrictive assumption on the exponential mixing rates and study the most basic assumptions that can be made on a process, while still performing statistical inference. We choose to study processes that are both stationary and ergodic (stationary–ergodic), where a *stationary* process only depends on the relative, and not absolute, position of observations, and *ergodic* in that statistical dependence vanishes asymptotically.

No Bootstrap algorithms exist for the general class of stationary–ergodic processes. Further, both variations on the block Bootstrap and Markov Bootstrap fail to capture the serial dependence structure in this general class without making restrictive exponential mixing assumptions. Here we propose a novel, principally different, approach to generating Bootstrap sequences, that is based on replacements. The *replacement Bootstrap* relies on estimating the conditional replacement distribution of a specific observation (symbol) in the sequence, given observations in its past and future. Rather than generating new sequences (bootstraps) from scratch, the replacement Bootstrap leverages the available sample by introducing changes (replacements) according to an *estimated* conditional replacement distribution. The intention is to preserve the underlying structure of the sample, while

introducing changes that conform to the structure of the process, observed from the sample sequence. The performance of the method depends on the number of replacement points R. First, R positions are randomly selected. Then, a replacement distribution is estimated on the R positions from the full observation sequence. And finally, R symbols are drawn simultaneously from the estimated replacement distribution to replace the R positions.

This chapter introduces an iterative replacement Bootstrap algorithm, $\mathcal{R}$-Boot, that estimates the replacement distribution according to the universal measure $\mathcal{R}$ [Ryabko, 1988, 2008]. Theoretical consistency guarantees for the *conditional (replacement)* distribution, that hold for arbitrary stationary–ergodic distributions, are proven, without relying on finite-memory or mixing conditions. Further, we empirically study the accuracy of the proposed method on a particularly challenging extrema statistic, the maximum drawdown (note that theoretical consistency results for this complex statistic are not studied). Theoretical results establish the consistency of the proposed method for generating new time-series and empirical results for the maximum drawdown are evaluated (note that empirical estimation performance for the mean and standard deviation are also included for completeness). To our knowledge, this is the first theoretically grounded attempt to use the Bootstrap for such a general class of stationary–ergodic processes.

The organization of the chapter is as follows. First, notation is set in Section 2. Then the nonparametric Bootstrap Principle, i.i.d. Bootstrap, block Bootstrap and Markov Bootstrap are formalized in 3. Section 4 introduces the replacement Bootstrap principle for dependent data, the iterative $\mathcal{R}$-Boot algorithm and consistency guarantees. Section 5 presents numerical results comparing $\mathcal{R}$-Boot with standard Bootstrap approaches on simulated and real dependent data, using a practically relevant and complex extrema statistic. Finally, Section 6 concludes the chapter and Section 7 presents future work.

2 Preliminaries

In this section we introduce the notation used throughout the chapter. A sequence $\boldsymbol{X}_T = (X_1, \ldots, X_T)$ is generated by a process P over a finite alphabet A, where the capital letter indicates that X_t is a random variable. Denote by $\boldsymbol{X}_{<t} = (X_1, \ldots, X_{t-1})$, $\boldsymbol{X}_{>t} = (X_{t+1}, \ldots, X_T)$, and $\boldsymbol{X}_{t:t'} = (X_t, \ldots, X_{t'})$, the past, future, and internal subsequences of $\boldsymbol{X}_T$. Note that $\boldsymbol{X}_{1:T} = (X_1, \ldots, X_T)$ can also be used to indicate the range from 1 to T. Finally, $\mathcal{U}([a, b])$ denotes a uniform distribution on the interval $[a, b]$.

We assume the following:

Assumption 1. *The process P is* stationary*, i.e., for any m, τ, and word $\boldsymbol{v}_m = (a_1, \ldots, a_m) \in A^m$,*

$$\mathbb{P}(X_1 = a_1, \ldots, X_m = a_m) = \mathbb{P}(X_{1+\tau} = a_1, \ldots, X_{m+\tau} = a_m),$$

and

Assumption 2. *The process P is* ergodic*, i.e., for any word $\boldsymbol{v}_m = (a_1, \ldots, a_m)$ the empirical frequency of $\boldsymbol{v}_m$ in a sequence $\boldsymbol{X}_T$ tends to its probability,*

$$\frac{\nu_{\boldsymbol{X}_T}(a_1, \ldots, a_m)}{T} \to \mathbb{P}(X_1 = a_1, \ldots, X_m = a_m), \ \ a.s.,$$

where

$$\nu_{\boldsymbol{X}_T}(a_1, \ldots, a_m) = \#\{s \leq T : X_s = a_m, \ldots, X_{s-m+1} = a_1\},$$

denotes the number of occurrences of word $\boldsymbol{v}_m$ in $\boldsymbol{X}_T$.

The latter definition of ergodicity is equivalent to the standard definition involving shift-invariant sets [Gray, 1988].

We recall the definition of the Kullback-Leibler (KL) divergence used to measure the accuracy of estimated distributions. Given two distributions P and Q

Input:

- Symbols $a \in A$
- Sample $\boldsymbol{X}_T$, of length T
- Bootstrap algorithm $\mathcal{B}_T : A^T \rightsquigarrow A^T$
- Number of Bootstraps B

For $i = 1, 2, \ldots, B$**, repeat**

$\boldsymbol{b}_T^i = \mathcal{B}_T(\boldsymbol{X}_T)$

end for

Figure 2.1: The Bootstrap Protocol

over A, the KL divergence between P and Q is defined as,

$$\mathrm{KL}(P;Q) = \sum_{a \in A} P(a) \log \frac{P(a)}{Q(a)}. \tag{2.1}$$

We also recall that the Shannon entropy of a random variable X distributed according to P is,

$$h(X) := -\sum_{a \in A} P(X = a) \log P(X = a),$$

with the convention that $0 \log 0 = 0$. In general, the k-order entropy of a process P is defined as,

$$h_k(P) := \mathbb{E}_{\boldsymbol{X}_{1:k}}[h(X_{k+1}|\boldsymbol{X}_{1:k})],$$

which is non-increasing with k. Since it is non-negative, the sequence $h_k(P)$, $k \in \mathcal{N}$ has a limit, which is denoted by $h_\infty(P)$ and is called the *entropy rate* of the time-series distribution P.

3 The Bootstrap

While estimating a statistic $\hat{f}$ on a limited sample sequence $\boldsymbol{X}_T$, nonparametric Bootstrap algorithms make no parametric assumptions on the process P, relying solely on the input sequence $\boldsymbol{X}_T$ to generate *bootstrap* sequences. Statistics of interest are then computed on these *bootstrap* sequences. A Bootstrap algorithm

Input:

- Symbols $a \in A$
- Sample $\boldsymbol{X}_T$, of length T

Compute: $\nu_{\boldsymbol{X}_T}(a) = \#\{t : X_t = a\}$
For $t = 1, 2, \ldots, T$, **repeat**

$b_t \sim \frac{\nu_{\boldsymbol{X}_T}(a)}{T}$

end for
Output: $\boldsymbol{b}_T$

Figure 2.2: i.i.d. Bootstrap Algorithm

$\mathcal{B}_T(\cdot)$ is a *random* mapping $\mathcal{B}_T : A^T \rightsquigarrow A^T$, such that given a sequence $\boldsymbol{X}_T$, $\mathcal{B}_T(\boldsymbol{X}_T)$ returns a (random) Bootstrap sequence $\boldsymbol{b}_T$ (a formal protocol is presented in Figure 2.1). The intuition is that the sampling distribution (based on a sample sequence $\boldsymbol{X}_T$) provides enough information to generate realistic sequences which are *likely* to have been generated by P. Consistency guarantees depend on the continuity of mapping the sampling distribution to the population distribution. As the latter distribution is unavailable, it is approximated using the Bootstrap distribution.

The original i.i.d. Bootstrap computes the symbol frequencies $\nu_{\boldsymbol{X}_T}(a) = \#\{t : X_t = a\}$, for any $a \in A$, placing probability mass $\frac{1}{T}$ on each observation, and generates Bootstrap sequences $\mathcal{B}_T^{\text{iid}}(\boldsymbol{X}_T) = \boldsymbol{b}_T$, such that $b_t \sim \frac{\nu_{\boldsymbol{X}_T}(a)}{T}$ (the algorithm is formally presented in Figure 2.2). Given B bootstrap sequences $\boldsymbol{b}_T^1, \ldots, \boldsymbol{b}_T^B$, the Bootstrap estimator $\tilde{\theta}_T = \frac{1}{B}\sum_{i=1}^{B} \hat{f}\left(\boldsymbol{b}_T^i\right)$ is likely closer to the statistic of interest $\theta_T = \mathbb{E}[\hat{f}(\boldsymbol{X}_T)]$, than by simply using the asymptotic estimator $\hat{\theta}_T = \hat{f}\left(\boldsymbol{X}_T\right)$. Convergence results require that $B \to \infty$, so that the Bootstrap estimator converges to the Bootstrap distribution. A common application area is in finance [Cogneau and Zakamouline, 2010], where an *a priori* analytic study or assumptions on either the process or statistic is not possible. Other applications include estimating distribution functions, residuals, generating confidence intervals or performing hypothesis tests.

Under certain circumstances, the original i.i.d. Bootstrap of Efron [1979] outperforms asymptotic estimators (based on the central limit theorem). When the es-

Input:

- Block width m
- Sample $\boldsymbol{X}_{1:T}$

For $t = 1, \ldots, \lceil \frac{T}{m} \rceil$, **repeat**

$u \sim \mathcal{U}([1, T - m + 1])$

$b_{(t-1)*m+1,\ldots,t*m} = X_{u,\ldots,u+m-1}$

end for
Output: $\mathbf{b} = b_{1,\ldots,T}$

Figure 2.3: Block Bootstrap Algorithm

timator is a smooth function of moments on the process, MacKinnon [2006] shows that the i.i.d. Bootstrap improves on the simple asymptotic estimator. Horowitz [2001] shows that finite sample improvements are only possible for asymptotically pivotal statistics, that is, statistics with an asymptotic distribution that does not depend on the unknown population parameters or form of the process. In the case of asymptotically pivotal statistics with symmetric distributions, the Bootstrap converges to the empirical cumulative distribution function (cdf)[1] at a rate of $\mathcal{O}\left(T^{-\frac{3}{2}}\right)$, while asymptotic estimates converge at a rate of $\mathcal{O}\left(T^{-\frac{1}{2}}\right)$. This is quite an advantage. Unfortunately, negative results have been shown for dependent sequences, extrema statistics, boundary parameters and several other distributions (for a review of negative results, see e.g. Horowitz [2001]). As a result, dependent data alternatives need to be considered when dealing with a dependent observation sequence.

3.1 Block Bootstrap

When applying the Bootstrap to time series, special handling of the sample data is required to retain the dependence structure and generate realistic variability. The most general nonparametric Bootstrap approach for time series is the block Bootstrap (see e.g., Figure 2.3, for a full review of block methods, see e.g., Kreiss and Lahiri [2012]). Block methods capture the dependence structure at lag dis-

[1]Note that anytime we report "convergence" results in this chapter, we are referring to the convergence to the empirical cdf of a symmetric probability distribution.

tances defined by a block width. These methods segment data into blocks to retain the in–block temporal dependency and generate bootstrap sequences by sampling blocks with replacement and joining them end-to-end. As the distribution of blocks implicitly models the dependency structure of the process, block selection probability can optionally be altered through "tilting" (weighting) probabilities [Hall and Yao, 2003] or by "matching" blocks according to their Markov transition probabilities [Carlstein et al., 1998]. Block construction methods include moving blocks [Kuensch, 1987, Liu and Singh, 1992], non-overlapping blocks [Carlstein, 1986], circular blocks [Politis and Romano, 1992], tapered blocks [Paparoditis and Politis, 2001], matched blocks [Carlstein et al., 1998] and stationary blocks [Politis and Romano, 1994] (for a relative performance overview, please see e.g., Lahiri [1999], Nordman et al. [2009]).

The process of segmenting data into blocks disrupts the dependency structure of the data, so estimator performance is very sensitive to block width. The choice of smaller than optimal block width increases the bias of the estimator, while selecting a larger than optimal block width increases its variance. The stationary Bootstrap reduces block sensitivity by drawing block widths according to a geometric distribution, but it is asymptotically inefficient compared to the circular block Bootstrap, which assumes data lie on a circle [Horowitz, 2001].

Consistency results for block Bootstrap estimators rely on asymptotically *optimal* block width selection, which is defined as the block width that minimizes the asymptotic mean-square error of the block Bootstrap estimator, and require that processes are both stationary and strongly exponentially mixing. Hall et al. [1995] derives optimal block selection rules as a function of the autocovariance function of the time series, showing that the (expected) block size should increase at the rate of $\mathcal{O}\left(T^{\frac{1}{3}}\right)$. Zvingelis [2003], Politis and White [2004], Patton et al. [2009] extend this to an automatic block width selection procedure (for the circular and stationary block Bootstrap algorithms) based on estimates for both the autocovariance and spectral density. Assuming that the *optimal* block width is chosen, then block Bootstrap estimators for symmetric probabilities converge a.s. $\mathcal{O}\left(T^{-\frac{6}{5}}\right)$ [Hall et al., 1995].

Input:

- Markov Model size k
- Symbols $a \in A$
- Sample $\boldsymbol{X}_{1:T}$

Compute:

- Frequencies $\nu_{\boldsymbol{X}_T}$,
$$\nu_{\boldsymbol{X}_T}(a) = \#\{t : X_t = a\}$$
- k-Markov patterns $\boldsymbol{v}_k = (v_1, \ldots, v_k) \in A^k$,
$$\nu_{\boldsymbol{X}_T}(a|\boldsymbol{v}_k) = \#\{s < T : X_s = a; X_{s-k} = v_1, .., X_{s-1} = v_k\}$$

For $i = 1, \ldots, k$**, repeat**

$$b_t \sim \frac{\nu_{\boldsymbol{X}_T}(a)}{\sum_{c \in A} \nu_{\boldsymbol{X}_T}(c) + |A|}$$

end for
For $i = k+1, 2, \ldots, T$**, repeat**

$$b_t \sim \frac{\nu_{\boldsymbol{X}_T}(a|\boldsymbol{v}_k)}{\sum_{c \in A} \nu_{\boldsymbol{X}_T}(c|\boldsymbol{v}_k) + |A|}$$

end for
Output: $\boldsymbol{b}_T$

Figure 2.4: Markov Bootstrap Algorithm

Block methods have also been shown to be inconsistent estimators of the means for some strongly dependent and long-range dependent processes [Lahiri, 1993], as well as heavy tails, distributions of the square of a sample average, distributions of the maximum of a sample and parameters on a boundary of the parameter space [Horowitz, 2001]. These methods *necessarily* ignore and *necessarily* break long-range dependence and fail under poor block width selection [Lahiri, 1999].

3.2 Markov Bootstrap

The Markov Bootstrap estimates the Markov transition density according to a (specified) k-order Markov model using any nonparametric Markov estimator. Bootstrap sequences are then sequentially generated by iteratively sampling T symbols from the estimated Markov model, which are conditioned on the previous

k symbols generated along the Bootstrap sequence. This procedure is referred to as "Markov conditional Bootstrap" and is not a replacement for the block Bootstrap, but an attractive alternative when in addition to the standard assumptions, the process has finite-memory and generated using a Markov model. Under these additional restrictive conditions, the Markov Bootstrap convergence rate (for symmetrical probabilities) is a.s. $\mathcal{O}\left(T^{-(3/2+\varepsilon)}\right)$, for any $\varepsilon > 0$ [Horowitz, 2003]. Accurately modeling the dependency of a process with the Markov Bootstrap requires correct model specification. Setting an incorrect model size results in complete failure Horowitz [2003], where setting it too small results in an estimator with too much bias and setting it too large results in an estimator with too much variance.

4 The Replacement Bootstrap

In this chapter, we consider a set of assumptions that are far more general than those considered in the block and Markov Bootstrap. In the case of short sequences, where data correlations do not fully decay (not strongly exponential mixing), or processes which are not finite memory, standard Bootstrap methods are not consistent. Our results allow us to consider this much larger class of processes. By considering the general class of stationary–ergodic processes, we do not require that the process reduces to a correct Markov model size or block width. Further, as mixing conditions do not hold for this general class, it is not possible to apply standard analysis methods, so we instead focus on studying the consistency of the estimated conditional replacement distribution of values in the observation sequence.

Here we introduce a novel approach to the Bootstrap for dependent sequences, where bootstraps sequences are generated by *replacing* symbols in the original sequence according to an estimate of their probabilities, conditioned on values observed both before and after each of the symbols, in the observed sequence. The *replacement* Bootstrap randomly selects a set of R points at positions $t_i \sim \mathcal{U}(\{1, \ldots, T\}), i = 1, \ldots, R$, in the original sequence $\boldsymbol{X}_T$, simultaneously replacing symbols $(X_{t_1}, \ldots, X_{t_R})$ with symbols $(b_{t_1}, \ldots, b_{t_R})$, ideally drawn from the

conditional distribution

$$\mathbb{P}(a_1, \ldots, a_R | \boldsymbol{X}_T \setminus \{X_{t_1}, X_{t_2}, \ldots, X_{t_R}\}).$$

As the conditional distribution is unavailable, it must be estimated. A few points need to be noted. First, in constructing bootstrap sequences according to the traditional Bootstrap principle, symbols (or blocks) are drawn according to their frequency distribution in the sample sequence. This is traditionally modeled by a block size or Markov model size. Here, symbols are drawn according to a conditional replacement distribution that additionally integrates information on their position in the sequence. This differs from existing techniques in that neighborhood information is restricted to a specific block width or model size, while the replacement Bootstrap leverages much more information from the observation sequence. As the number of replacements R increases, the variability of the generated bootstrap sequence grows due to an increasing bias from errors in the conditional probability estimates. Each increase in R, increases the degrees of freedom for possible bootstrap sequences, defining how much of the original sequence is preserved. Consequently, small R increases estimator bias and large R increases variance. This replacement strategy simultaneously exploits the *full* sequence to determine symbol replacements, while partially preserving the temporal structure of the original sample observation sequence.

This replacement based approach depends on an accurate estimate of the *conditional replacement distribution*. This is unknown by the nature of the setting. To gain an intuition for the conditional replacement distribution, let us consider the simpler case of $R = 1$, i.e., replacing a single symbol in a single random location t. While X_t could simply be replaced with the estimated conditional distribution $\mathbb{P}(a|\boldsymbol{X}_{<t})$, the *prediction estimate*, this only uses the *past* information to compute the conditional probability. Note that the Markov Bootstrap leverages a prediction estimate based on a single k-order Markov model, estimated from the sample series, and only conditions on the previous k generated symbols while generating a Bootstrap sequence. Further, unlike the Markov Bootstrap, the replacement

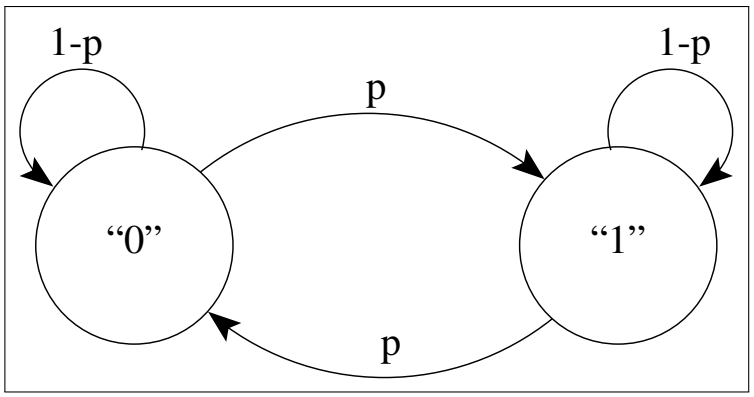

Figure 2.5: A simple 1-Markov chain.

Bootstrap does not assume or require the explicit specification of a model size.

The conditional replacement probability used in the replacement Bootstrap is estimated on the whole sequence, except the portion which is to be replaced, e.g. $\mathbb{P}(a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$. Note that this reasoning holds for any value of R, where the conditional replacement probability for any value of $R \geq 1$ is simply $\mathbb{P}(a_1, \ldots, a_R|\boldsymbol{X}_T \setminus \{X_{t_1}, \ldots, X_{t_R}\})$. This procedure results in greater estimation accuracy over the symbol replacement distribution at X_t, capturing as much of the dependency structure as possible from the sample observation sequence.

In general, it is possible to construct examples where the entropy of the process conditioned on $\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}$ tends to zero as the length of the *past* $\boldsymbol{X}_{<t}$ and *future* $\boldsymbol{X}_{>t}$ tends to infinity, while the process itself has a non-zero entropy. Thus, this results in a more accurate estimate of the conditional replacement distribution for symbols by better reproducing the unknown dependency structure of the process.

4.1 Comparing the Replacement Probability to the Prediction Probability

We now present a simple example illustrating the potential advantage of estimating the *replacement* distribution, rather than the *prediction* distribution. As an illustration of this advantage, consider the simple case of replacing one symbol ($R = 1$) in a series generated by a 2-state Markov chain, as represented in Figure 2.5.

Let P be the 2-state Markov chain with an initial distribution equally dis-

tributed over state 0 and 1 and parameter $0 \leq p \leq 0.5$. Given a sequence $\boldsymbol{X}_T$ generated from P, we want to replace a symbol at t. In particular, we consider a sequence where,

$$(\ldots, X_{t-1}, X_t, X_{t+1}, \ldots) = (\ldots 010 \ldots).$$

Following an approach similar to the i.i.d. Bootstrap, we simply replace X_t with a new symbol b_t drawn from the empirical frequency $\frac{\nu_{\boldsymbol{X}_T}(\cdot)}{T}$ which tends to converge to 0.5 (the probability of 0 and 1). On the other hand, we could use a prediction approach generating b_t from the conditional distribution $\mathbb{P}(\cdot|\boldsymbol{X}_{<t})$. Since P is 1-Markov,

$$\mathbb{P}(\cdot|\boldsymbol{X}_{<t}) = \mathbb{P}(\cdot|X_{t-1}),$$

and we have,

$$\mathbb{P}(0|X_{t-1}) = 1 - p,$$

and

$$\mathbb{P}(1|X_{t-1}) = p.$$

Finally, the replacement distribution conditioned on both past and future around X_t results in,

$$\mathbb{P}(0|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) = \mathbb{P}(0|X_{t-1}, X_{t+1}) = \frac{(1-p)^2}{(1-p)^2 + p^2}.$$

It is easy to see that the entropy of the replacement distribution is much smaller than for prediction and i.i.d. distributions, implying that replacing X_t with b_t, drawn from the replacement distribution, is much more accurate than for these other approaches. For instance, for $p = 0.1$ we have,

$$h_{\text{iid}}(b_t) = 1,$$

$$h_{\text{prediction}}(b_t) = 0.469,$$

and

$$h_{\text{replacement}}(b_t) = 0.095.$$

□

4.2 The $\mathcal{R}$-Boot Algorithm

A direct implementation of the replacement Bootstrap requires an estimate of the probability,

$$\mathbb{P}(a_1, \ldots, a_R | \boldsymbol{X}_T \setminus \{X_{t_1}, X_{t_2}, \ldots, X_{t_R}\}),$$

which corresponds to the probability of a word $\boldsymbol{v} = (a_1, \ldots, a_R)$ in R random locations $t_1, \ldots, t_R$, conditioned on the remainder of the sequence, i.e., the portion of the sequence that is not replaced. Unfortunately, this requires estimating probabilities for an alphabet of size $|A|^R$ conditioned on different subsets of the sequence, which would rapidly become infeasible as R increases. Therefore, we propose a sequential process based on the estimation of the one-symbol replacement probability,

$$\mathbb{P}(\cdot | \boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}),$$

which results in a feasible and much more efficient procedure. Although a variety of methods can be used to estimate the conditional distribution, the algorithm presented here, $\mathcal{R}$-Boot, adapts a universal predictor (see e.g., Ryabko [2010]) to estimate the one-symbol conditional replacement distribution. Relying on a universal predictor to compute a consistent estimate of $\mathbb{P}(\cdot | \boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$ allows us to consider the wide class of stationary–ergodic processes and avoid restrictive model and/or parametric assumptions on the process P.

Definition 1. *A measure ρ is a universal predictor if for any stationary and ergodic process P,*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{X}_{<t}} \Big[KL\big(\mathbb{P}(\cdot | \boldsymbol{X}_{<t}); \rho(\cdot | \boldsymbol{X}_{<t})\big) \Big] \to 0, \tag{2.2}$$

where the expectation is w.r.t. P.

Several predictors with this property are known. Although a predictor ρ could be directly used to replace a symbol X_t by drawing a new symbol from the *pre-*

diction distribution $\rho(\cdot|\boldsymbol{X}_{<t})$, we show that a universal predictor can be adapted to estimate the conditional distribution $\mathbb{P}(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$. Thus, more accurate replacements can be achieved by taking into consideration not only the *past* $\boldsymbol{X}_{<t}$, but also the *future* $\boldsymbol{X}_{>t}$.

$\mathcal{R}$-Boot builds on the universal measure $\mathcal{R}$ [Ryabko, 1988, 2008], which unlike other universal predictors (e.g., the Ornstein predictor [Ornstein, 1978]) or compression-based predictors (e.g., Ryabko and Monarev [2005], Ryabko [2009]), is both resource and data efficient. It is defined as a combination of varying order Krichevsky predictors [Krichevsky, 1968] that only consider $\boldsymbol{X}_{<t}$ (e.g., the past) to estimate the prediction probability $\mathbb{P}(\cdot|\boldsymbol{X}_{<t})$ and is defined in Definition 2.

Definition 2. *For any $m \geq 0$, the Krichevsky predictor of order m estimates $\mathbb{P}(X_t = a|\boldsymbol{X}_{<t})$ as,*

$$\mathcal{K}^m(X_t = a|X_{t-m} = v_1, \ldots, X_{t-1} = v_m) = \begin{cases} \frac{\nu_{\boldsymbol{X}_T}(a|\boldsymbol{v}_m)+\frac{1}{2}}{\sum_{c\in A}\nu_{\boldsymbol{X}_T}(c|\boldsymbol{v}_m)+\frac{|A|}{2}}, & t > m \\ \frac{1}{|A|}, & t \leq m \end{cases}$$

where word,

$$\boldsymbol{v}_m = (v_1, \ldots, v_m) \in A^m,$$

contains the m symbols up to $t-1$ observed in $\boldsymbol{X}_T$, and $\nu_{\boldsymbol{X}_T}$ represents the word count in $\boldsymbol{X}_T$ up to $t-1$, that is for any $\boldsymbol{v}_m$ and a,

$$\nu_{\boldsymbol{X}_T}(a|\boldsymbol{v}_m) = \#\{s < t : X_s = a; X_{s-m} = v_1, .., X_{s-1} = v_m\},$$

with the convention that if $m = 0$, then $v = \emptyset$ and $\nu_{\boldsymbol{X}_T}(a|\emptyset) = \nu_{\boldsymbol{X}_T}(a) = \#\{s < t : X_s = a\}$.

Additive factors $\frac{1}{2}$ and $\frac{|A|}{2}$ make the Krichevsky predictor minimax optimal for any fixed sequence length and set of Markov sources, when the error is measured with the expected KL divergence. For sake of reference, notice that the additive

factors in the Laplace predictor are simply 1 and $|A|$, resulting in the predictor

$$\mathcal{L}^m(X_t = a|X_{t-m} = v_1, \ldots, X_{t-1} = v_m) = \begin{cases} \frac{\nu_{\boldsymbol{X}_T}(a|\boldsymbol{v}_m)+1}{\sum_{c\in A}\nu_{\boldsymbol{X}_T}(c|\boldsymbol{v}_m)+|A|}, & t > m \\ \frac{1}{|A|}, & t \leq m \end{cases}$$

which is not minimax optimal. While other divergences would give different predictors, the KL divergence is a natural choice for this problem. Therefore, it is used in our theoretical analysis. From the conditional distribution in Definition 2, the predictor $\mathcal{K}^m(\boldsymbol{X}_{1:T})$ is computed as,

$$\mathcal{K}^m(\boldsymbol{X}_{1:T}) = \prod_{t=1}^{T} \mathcal{K}^m(X_t|\boldsymbol{X}_{t-m:t-1}).$$

The measure $\mathcal{R}$ is then defined as follows.

Definition 3 (Ryabko [1988]). *For any t, the measure $\mathcal{R}$ is defined as,*

$$\mathcal{R}(\boldsymbol{X}_{1:T}) = \sum_{m=0}^{\infty} \omega_{m+1}\mathcal{K}^m(\boldsymbol{X}_{1:T}), \tag{2.3}$$

with weights,

$$\omega_m = (\log(m+1))^{-1} - (\log(m+2))^{-1}.$$

Thus, the measure $\mathcal{R}$ is an estimator constructed directly from $\boldsymbol{X}_T$ and does not rely on any parametric assumptions on the process. As proved in Ryabko [1988], $\mathcal{R}$ is a consistent universal estimator of the conditional probability for any stationary–ergodic process (see Definition 1) as $T \to \infty$.

In this novel replacement Bootstrap setting, the whole sequence $\boldsymbol{X}_T$ is used to compute the counters $\nu_{\boldsymbol{X}_T}$. This is in contrast to the standard Krichevsky predictor that does not consider the "future" of the sequence. Here we propose the following method of using $\mathcal{R}$ to generate Bootstrap sequences. Let $t \leq T$ be an arbitrary point in the original sequence. We replace the original symbol X_t with a new symbol b_t drawn from the conditional replacement distribution, i.e.,

Input:

- Sequence $\boldsymbol{X}_T$
- Replacements R
- Maximum pattern size K_T

Set $\boldsymbol{b}_T^0 = \boldsymbol{X}_T$
For all $m = 0, \dots, K_T$**, repeat**

Compute counts $\nu_{\boldsymbol{X}_T}(a|\boldsymbol{v}_m)$ for any $a \in A$, $\boldsymbol{v}_m \in A^m$

end for
For all $r = 1, \dots, R$**, repeat**

1. Draw the replacement point $t_r \sim \mathcal{U}([1, T])$
2. Draw $b_{t_r} \sim \mathcal{R}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{b}_{<t_r}^{r-1}; \boldsymbol{b}_{>t_r}^{r-1})$
3. Set $\boldsymbol{b}_T^r = (\boldsymbol{b}_{<t_r}^{r-1}, b_{t_r}, \boldsymbol{b}_{>t_r}^{r-1})$

end for

Figure 2.6: Pseudo-code of the $\mathcal{R}$-Boot algorithm.

$\mathbb{P}(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$, estimated using the $\mathcal{R}$ measure,

$$\mathcal{R}_{\boldsymbol{X}_T}(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) = \frac{\mathcal{R}_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c \in A} \mathcal{R}_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}. \tag{2.4}$$

Once the conditional replacement probability is estimated, $\mathcal{R}$-Boot substitutes X_t in the original sequence with b_t drawn from $\mathcal{R}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$, thus obtaining the new sequence $\boldsymbol{z}_T^1 = (\boldsymbol{X}_{<t}, b_t, \boldsymbol{X}_{>t})$. Here $\mathcal{R}_{\boldsymbol{X}_T}$ (resp. $\mathcal{K}^i_{\boldsymbol{X}_T}$) refers to the measure $\mathcal{R}$ (resp. Krichevsky predictor) with frequency counts $\nu(\cdot)$ from Definition 2, computed only once from the original sequence $\boldsymbol{X}_T$. Once X_t is replaced by b_t, resulting in $\boldsymbol{z}_T^1$, the counts are not recomputed on $\boldsymbol{z}_T^1$. $\mathcal{R}_{\boldsymbol{X}_T}$ is used as an estimate of the conditional replacement distribution for all subsequent replacements. This process is iterated R times, such that, at each replacement r, a random point t_r is chosen and the new symbol b_{t_r} is drawn from $\mathcal{R}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{z}_{<t_r}^{r-1}, \boldsymbol{z}_{>t_r}^{r-1})$. Finally, the sequence $\boldsymbol{b}_T = \boldsymbol{z}_T^R$ is returned. The pseudo-code of $\mathcal{R}$-Boot is reported in Figure 2.6.

Briefly, the idea of $\mathcal{R}$-Boot is to iteratively replace a single random position, in the current iteration of the Bootstrap sequence, until R points have been replaced.

Estimating the conditional distribution at each iteration translates to estimating the conditional distribution for each of the possible symbols $a \in A$ for that position in the sequence. The replacement symbol is then drawn according to this estimated replacement distribution to replace the current symbol in that position.

4.3 Impact of Replacement Parameter R

The replacement parameter R has the same expected impact on iterative replacements as it has on simultaneous replacements. The number of replacements R defines how much of the original sequence is preserved, where small R increases bias for the original sequence and large R favors greater variance. In particular, the temporal structure of the original sequence is preserved for smaller values of R, while larger values of R increase variability and noise. In general, the incremental nature of $\mathcal{R}$-Boot requires large R (generally greater than T) to compensate for the iterative (versus simultaneous) nature of replacements. Though unlikely, the random selection of replacement points might result in repeated selection of the same position in the series. This can cause problems in that the aim is to cover the series and not to localize replacements to a single neighborhood. Conversely, we found that some repeated selection actually improved performance by introducing variability through the iterative process. For example, for X_{t_r}, at step r, $\mathcal{R}$-Boot uses the current sequence $\boldsymbol{z}_T^{r-1}$ to define the conditional probability $\mathcal{R}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{z}_T^{r-1}, \boldsymbol{z}_T^{r-1})$. As a result, changes to the symbols $X_{t_1}, \ldots, X_{t_{r-1}}$ in the original sequence may later trigger changes in other locations and this source of variability was a positive contributer to good performance. In testing, we found that removing duplicate replacement points caused a noticeable reduction in performance. As a result, it is important to retain random selection, while taking care to notice potential issues with small values of R.

4.4 Implementation details

This section simplifies Equation 2.4 to show how the conditional replacement distribution can be efficiently computed. Combining an infinite number of Krichevsky

predictors in the measure $\mathcal{R}$ can be computed in polynomial time by setting the maximum Krichevsky model size to $K_T = \mathcal{O}(\log T)$. Estimates of order m only uses m samples in the past, so this considerably reduces the number of computations needed to estimate the measure $\mathcal{R}$. This results in R replacements of order $\mathcal{O}\left((T + R\log T)\log^2(T)|A|^2\right)$ to generate a single Bootstrap sequence.

First note that replacing infinity with any K_T increasing to infinity with T does not affect the asymptotic convergence properties of the measure $\mathcal{R}$. Next, frequency estimates in $\mathcal{K}^m$ for $m \gg O(\log T)$ are not consistent, so they only add noise to the estimate. Therefore, setting K_T to $O(\log T)$ is meaningful, while also efficiently computable, in that it retains meaningful structure, while also significantly reducing computational complexity.

We begin by first elaborating Equation 2.4. First, replacement points are drawn at random from a uniform distribution, i.e., $t \sim \mathcal{U}([1, T])$. The probability that a (new) sequence has symbol $a \in A$ in position t given that the rest of the sequence is equal to the original sequence $\boldsymbol{X}_T$ is,

$$\mathbb{P}(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}),$$

and is estimated with measure $\mathcal{R}$ as,

$$\begin{aligned}
\mathcal{R}_{\boldsymbol{X}_T}(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) &= \frac{\mathcal{R}_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A}\mathcal{R}_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})} && (2.5)\\
&= \frac{\sum_{i=0}^{\infty}\omega_i\mathcal{K}^i_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A}\sum_{j=0}^{\infty}\omega_j\mathcal{K}^j_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}\\
&= \sum_{i=0}^{\infty}\frac{\omega_i\mathcal{K}^i_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A}\sum_{j=0}^{\infty}w_j\mathcal{K}^j_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}\\
&= \sum_{i=0}^{\infty}\left[\sum_{c\in A}\sum_{j=0}^{\infty}\frac{\omega_j}{\omega_i}\frac{\mathcal{K}^j_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}{\mathcal{K}^i_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}\right]^{-1}. && (2.6)
\end{aligned}$$

Although the previous expression could be computed directly by using the definition of the Krichevsky predictors, the values returned by $\mathcal{K}^j_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})$ and $\mathcal{K}^i_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})$ would rapidly fall below the precision threshold, thus introducing significant numerical errors. We reformulate the previous expression to

manage this problem.

For any $i, j \neq 0$, let $y^c = \{\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t}\}$, $y^a = \{\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t}\}$ and $t'_{i,j} = t + \max\{i, j\}$, then,

$$\begin{aligned}
\frac{\mathcal{K}^j_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}{\mathcal{K}^i_{\boldsymbol{X}_T}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})} &= \prod_{s=1}^{T} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{<s})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{<s})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{<s})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{<s})} \prod_{s=t}^{T} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{<s})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{<s})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_s|X_{<s})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_s|X_{<s})} \prod_{s=t}^{T} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{<s})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{<s})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-j:s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-i:s-1})} \prod_{s=t}^{T} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{s-j}, \ldots, y^c_{s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{s-i}, \ldots, y^a_{s-1})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-j:s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-i:s-1})} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{s-j}, \ldots, y^c_{s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{s-i}, \ldots, y^a_{s-1})} \prod_{s=t'_{i,j}+1}^{T} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{s-j}, \ldots, y^c_{s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{s-i}, \ldots, y^a_{s-1})}
\end{aligned}$$

$$\begin{aligned}
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-j:s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-i:s-1})} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{s-j}, \ldots, y^c_{s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{s-i}, \ldots, y^a_{s-1})} \prod_{s=t'_{i,j}+1}^{T} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-j:s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-i:s-1})} \\
&= \pi_{i,j,1:t-1} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y^c_s|y^c_{s-j}, \ldots, y^c_{s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s|y^a_{s-i}, \ldots, y^a_{s-1})} \pi_{i,j,t'_{i,j}+1:T},
\end{aligned}$$

where for any t_1 and t_2, we define,

$$\pi_{i,j,t_1:t_2} = \prod_{s=t_1}^{t_2} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-j:s-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_s|\boldsymbol{X}_{s-i:s-1})}.$$

Finally, from the definition of π, we have that computations can occur in either direction using the following recursive equations,

$$\begin{aligned}
\pi_{i,j,t_1:t_2+1} &= \pi_{i,j,t_1:t_2} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_{t_2}|X_{t_2-j}, \ldots, X_{t_2-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_{t_2}|X_{t_2-i}, \ldots, X_{t_2-1})}, \\
\pi_{i,j,t_1:t_2} &= \pi_{i,j,t_1+1:t_2} \frac{\mathcal{K}^j_{\boldsymbol{X}_T}(X_{t_1}|X_{t_1-j}, \ldots, X_{t_1-1})}{\mathcal{K}^i_{\boldsymbol{X}_T}(X_{t_1}|X_{t_1-i}, \ldots, X_{t_1-1})}.
\end{aligned}$$

The previous expression is still valid for either $i = 0$ or $j = 0$ with the conven-

tion that $\mathcal{K}^0(y_s^b|y_{<s}^b) = \mathcal{K}^0(y_s^b)$. Furthermore, for any t_1 and t_2, when $i = j$, the corresponding $\pi_{i,j,t_1:t_2} = 1$, while for any $i \neq j$ $\pi_{i,j,t_1:t_2} = \pi_{j,i,t_1:t_2}^{-1}$, thus the number of computations is halved for i and j. We are left with computing the expression $\mathcal{K}^i_{\boldsymbol{X}_T}(y_s|y_{s-i},\ldots,y_{s-1})$ (for $y = y^a$ and $y = y^c$) over the sequence of values dependent on the replacement at t. Let $y_s = a$ and $y_{s-i},\ldots,y_{s-1} = v^i$, where v^i is some word of length i, then,

$$\begin{aligned}\mathcal{K}^i_{\boldsymbol{X}_T}(y_s|y_{s-i},\ldots,y_{s-1}) &= \frac{\nu_{\boldsymbol{X}_T}(a|v^i) + \frac{1}{2}}{\sum_{c\in A}\nu_{\boldsymbol{X}_T}(c|v^i) + \frac{|A|}{2}} \\ &= \frac{\nu_{\boldsymbol{X}_T}(a|v^i) + \frac{1}{2}}{\nu_{\boldsymbol{X}_T}(v^i) + \frac{|A|}{2}}.\end{aligned}$$

Each term is computable since they are only as small as 2^{-i}, so the ratios for each s and their product continue to be well conditioned. Even in the case that for some pairs i and j the numbers become too small, they are just added (in the summation over c and j), so this no longer poses any numerical approximation errors. This results in the following optimized computation of the measure $\mathcal{R}$,

$$\begin{aligned}&\mathcal{R}_{\boldsymbol{X}_T}(X_t = a|\boldsymbol{X}_{<t},\boldsymbol{X}_{>t}) = \\ &\sum_{i=0}^{K_T}\left[\sum_{j=0}^{K_T}\frac{\omega_j}{\omega_i}\pi_{i,j,1:t-1}\pi_{i,j,t'_{i,j}+1:T}\left(\sum_{c\in A}\prod_{s=t}^{t'_{i,j}}\frac{\mathcal{K}^j_{\boldsymbol{X}_T}(y_s^c|y_{s-j}^c,\ldots,y_{s-1}^c)}{\mathcal{K}^i_{\boldsymbol{X}_T}(y_s^a|y_{s-i}^a,\ldots,y_{s-1}^a)}\right)\right]^{-1}.\end{aligned} \tag{2.7}$$

Next, we notice significant improvements in computational complexity can be achieved by storing repeated computations. More specifically, initializing counts ν and coefficients π, over alphabet A, to compute the first replacement distribution $\mathcal{R}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{X}_{<t},\boldsymbol{X}_{>t})$, requires a full scan through the sequence $\boldsymbol{X}_T$ and has a complexity of $O(TK_T^2A^2)$. By storing repeated calculations overlapping between replacements and only updating when necessary, the computational complexity of subsequent replacements can be dramatically reduced. In fact, only the coefficients $\pi_{i,j,t_1:t_2}$ are directly affected by replacing X_t by b_t. Therefore, it is possible to compute all subsequent replacements for $\mathcal{R} > 1$ with complexity $O(K_T^3A^2)$ by simply updating $\pi_{i,j,t_1:t_2}$ and recycling repeated calculations. The following structures are introduced to improve computational complexity. Let K be a matrix of

dimension $(K_T + 1) \times T$, such that for any $i = 0, \ldots, K_T$ and $s = 1, \ldots, T$,

$$K_{i,s} = \mathcal{K}^i_{\boldsymbol{X}_T}(X_s | \boldsymbol{X}_{s-i:s-1}).$$

We define the multi-dimensional matrix R, of dimensions $(K_T+1) \times (K_T+1) \times T$, such that for any $i, j = 0, \ldots, K_T$ and $s = 1, \ldots, T$,

$$R_{i,j,s} = \frac{K_{j,s}}{K_{i,s}}.$$

Then we compute π_{prev} and π_{post} as,

$$\pi_{i,j,\text{prev}} = \prod_{s=1}^{t-1} R_{i,j,s}$$
$$\pi_{i,j,\text{post}} = \prod_{s=t'_{i,j}+1}^{T} R_{i,j,s}.$$

Similarly we define the multi-dimensional matrix K', of dimensions,

$$(K_T + 1) \times |A| \times (t_{\text{end}} - t),$$

where $t_{\text{end}} = \min\{t'_{i,j}, T\}$, and,

$$K_{i,a,s} = \mathcal{K}^i_{\boldsymbol{X}_T}(y^a_s | y^a_{s-i}, \ldots, y^a_{s-1}).$$

Thus the matrix R', of dimensions,

$$(K_T + 1) \times (K_T + 1) \times |A| \times |A| \times (t_{\text{end}} - t),$$

is,

$$R'_{i,j,a,b,s} = \frac{K_{j,b,s}}{K_{i,a,s}}.$$

Once these structures are computed, the final probability of replacement for a

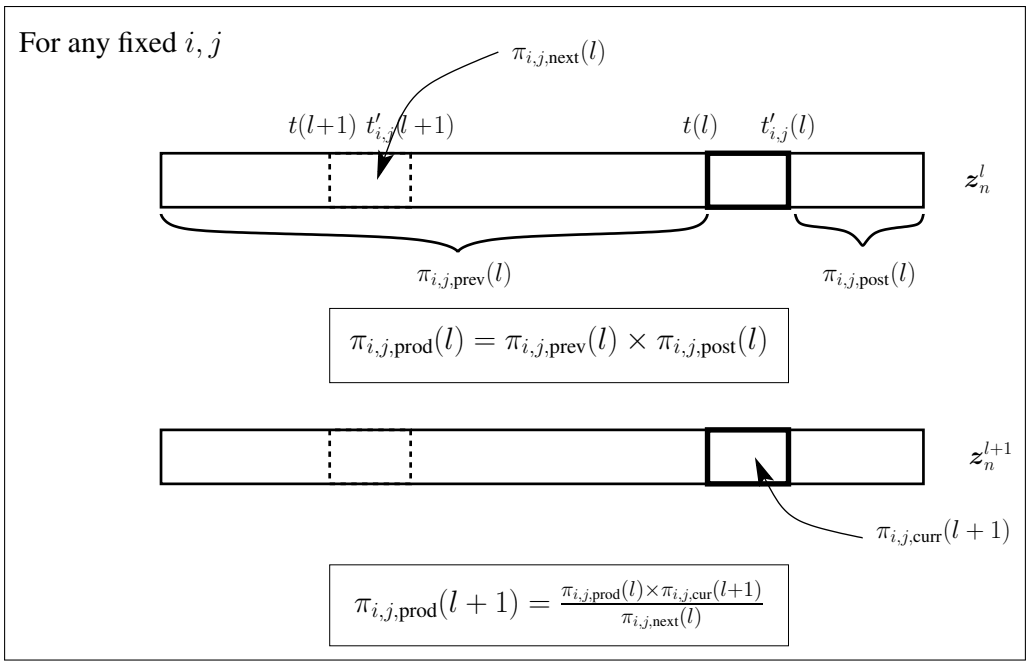

Figure 2.7: Computation of other replacement points (i.e., $l \geq 1$).

letter a in position t is efficiently computed as,

$$\mathcal{R}(X_t = a | \boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) = \sum_{i=0}^{K_T} \left[\sum_{j=0}^{K_T} \frac{w_j}{w_i} \pi_{i,j,\text{prev}} \pi_{i,j,\text{post}} \left(\sum_{b \in A} \prod_{s=t}^{t'_{i,j}} R'_{i,j,a,b,s} \right) \right]^{-1}. \quad (2.8)$$

We now use $l = 1, \ldots, L$ as an index for the replacement points; $t(l)$ as the time index of the l-th replacement point, $\boldsymbol{z}_T(l)$ as the sequence obtained after l replacement points (i.e., $\boldsymbol{z}_T(0) = \boldsymbol{X}_T$), and $\pi_{i,j,\text{prev}}(l)$ and $\pi_{i,j,\text{post}}(l)$ as the products of probabilities computed on the l-th sequence. Furthermore, we assume that the position of the replacement points is known in advance (i.e., $t(l)$ is chosen for all $l = 1, \ldots, L$ at the beginning). Following the efficient computation of the initial replacement $l = 1$, where the structures K, R, K', R' must be computed from scratch, we notice that for any $i = 1, \ldots, K_T$, and for any s such that $s < t(l)$, or $s > t(l) + i$, then $K_{i,s}(l+1) = K_{i,s}(l)$. Thus, for any i, only $K_{i,s}$ must be recomputed for $t(l) \leq s \leq t(l) + i$, which correspond to $O(K_T^2)$ updates. At the

first iteration $l = 0$, we compute

$$\pi_{i,j,\text{prev}}(0) = \prod_{s=1}^{t-1} R_{i,j,s}(0), \quad \pi_{i,j,\text{post}}(0) = \prod_{s=t'_{i,j}+1}^{T} R_{i,j,s}(0),$$

and the additional structures

$$\pi_{i,j,\text{prod}}(0) = \pi_{i,j,\text{prev}}(0) \times \pi_{i,j,\text{post}}(0),$$
$$\pi_{i,j,\text{next}}(0) = \prod_{s=t(1)}^{t'_{i,j}(1)} R_{i,j,s}(0).$$

Once the replacement is completed and $\boldsymbol{z}_T^1$ is computed, the $R_{i,j,s}$ values affected by the change are recomputed (updated) and used at the beginning of the new iteration to compute,

$$\pi_{i,j,\text{cur}}(1) = \prod_{s=t(0)}^{t'_{i,j}(0)} R_{i,j,s}(1),$$

which allows computation of the π_{prod} terms for the next iteration as,

$$\pi_{i,j,\text{prod}}(1) = \frac{\pi_{i,j,\text{prod}}(0)\pi_{i,j,\text{cur}}(1)}{\pi_{i,j,\text{next}}(0)}.$$

In general, after the first iteration, at any iteration $l \geq 1$, the computation of the previous elements can be iteratively updated as,

$$\pi_{i,j,\text{prod}}(l) = \frac{\pi_{i,j,\text{prod}}(l-1)\pi_{i,j,\text{cur}}(l)}{\pi_{i,j,\text{next}}(l-1)},$$

where,

$$\pi_{i,j,\text{next}}(l) = \prod_{s=t(l+1)}^{t'_{i,j}(l+1)} R_{i,j,s}(l), \quad \pi_{i,j,\text{cur}}(l) = \prod_{s=t(l-1)}^{t'_{i,j}(l-1)} R_{i,j,s}(l).$$

Finally, the computational complexity of $\mathcal{R}$-Boot to generate a single Bootstrap

with R replacements is $O((T + R \log T) \log^2(T) A^2)$.

4.5 A Comparison with Markov Bootstrap

While the replacement Bootstrap is very different from the block Bootstrap and does not assume the Markov property, it does share similarities with the Markov Bootstrap. For example, the $\mathcal{R}$-Boot algorithm uses Krichevsky predictors to estimate k-order Markov approximations of the process for $k = 1, \ldots, O(\log T)$. The measure $\mathcal{R}$ is then used to create a weighted combination of the K_T predictors, implicitly adapted on the prediction performance of each predictor. The Krichevsky predictor only considers the past k values in estimating the prediction probability $\mathbb{P}(\cdot|\boldsymbol{X}_{<t})$. More specifically, recall that the k-order Krichevsky predictor is computed as,

$$\mathcal{K}^k(\boldsymbol{X}_{1:T}) = \prod_{t=1}^{T} \mathcal{K}^k(X_t|\boldsymbol{X}_{t-k:t-1}).$$

First, this is limited to the k-order estimate of the model size. Second, this is quite limiting in comparison to the measure $\mathcal{R}$, which we recall is estimated as[2],

$$\mathcal{R}(\boldsymbol{X}_{1:T}) = \sum_{k=0}^{\infty} \omega_{k+1} \prod_{t=1}^{T} \mathcal{K}^k(X_t|\boldsymbol{X}_{t-k:t-1}).$$

Finally, the Markov Bootstrap is limited to probabilities drawn *only* from estimates conditioned on a k-order model that only conditions predictions on the k previous predictions. Whereas $\mathcal{R}$-Boot fully leverages all the available information both before and after each replacement by using an estimate of the replacement probability, $\mathbb{P}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$. $\mathcal{R}$-Boot *can* be viewed as a combination of k-order Markov estimators, but this is a severely limited perspective. Unlike the Markov Bootstrap, $\mathcal{R}$-Boot does not assume the Markov property, finite–memory or strongly exponential mixing times. The general space of stationary–ergodic processes is so rich and so much larger than the space of finite-memory processes, one could not expect to simulate any stationary–ergodic measure with some (even

[2] k is used in place of m to illustrate the similarity.

the best) k-order Markov measure. Next, as explained, the replacement Bootstrap does not generate the whole sequence based on the estimated distribution. It is more "conservative" (with regard to data) in that it retains the original sequence, only (sequentially) changing R symbols, selected at random, and based on the estimated replacement distribution. In fact, the measure $\mathcal{R}$ could be used to generate bootstrap sequences from scratch to directly generalize the Markov Bootstrap to the case of stationary–ergodic processes (which to our knowledge is a contribution in itself), but this would still only utilize the estimated prediction probability, and not the conditional replacement probability, which does not exploit the given sample sequence as effectively.

4.6 Theoretical Guarantees

A desirable property for an effective and consistent Bootstrap method is to produce an accurate distributional estimate from a finite sample sequence (i.e., $\mathbb{P}(\boldsymbol{X}_T)$). Unfortunately, this is not possible in the case of stationary–ergodic processes. This is in stark contrast to the classes of processes considered in existing Bootstrap algorithms, where estimating $\mathbb{P}(\boldsymbol{X}_{1:m})$ for a critical $m \ll T$ (e.g., $m = k+1$ in the case of k-order Markov processes) results in a sufficiently accurate estimate of $\mathbb{P}(\boldsymbol{X}_{1:T})$. While considering the general case of stationary–ergodic distributions significantly increases the applicability of the Bootstrap, it also prevents theoretical guarantees on the Bootstrap distribution estimate $\mathbb{P}(\boldsymbol{X}_{1:T})$. Moreover, it is provably impossible to establish any nontrivial rates of convergence for stationary–ergodic processes. Consequently, the following analysis will focus on asymptotic consistency guarantees for the individual replacements step in the $\mathcal{R}$-Boot Bootstrap algorithm.

At a high level, if the sequence length of at least one side of the replacement is sufficiently long, then, on average, the probability distribution for the inserted symbol converges to the true unknown probability distribution, of the symbol in that position, given the observations on both sides of the replacement, herein referred to as the past and future. Moreover, as the length of the sequence, both in the past and in the future, with respect to the replacement goes to infinity, the

probability distribution over symbols approaches the double-sided entropy rate.

Here, we introduce additional notation. A stationary distribution over one-way infinite sequences $X_1, X_2, \ldots$ can be uniquely extended to a distribution over two-way infinite sequences $\ldots, X_{-1}, X_0, X_1, \ldots$. We assume this extension whenever necessary. Recall that for stationary processes, the k-order entropy rate can be written as,

$$h_k(P) = \mathbb{E}_{\boldsymbol{X}_{-k:-1}}[h(X_0|\boldsymbol{X}_{-k:-1})].$$

Similar to the *entropy rate*, one can define the two-sided entropy,

$$h_{k,m}(P) = \mathbb{E}_{\boldsymbol{X}_{-k:-1},\boldsymbol{X}_{1:m}} h(X_0|\boldsymbol{X}_{-k:-1}, \boldsymbol{X}_{1:m}),$$

which is non-decreasing with k and m, and whose limit $\lim_{k,m\to\infty} h_{k,m}$ we denote by $h_{\infty\times\infty}$, where $\lim_{k,m\to\infty}$ is an abbreviation for "for every pair of increasing sequences $(k_l)_{l\in\mathcal{N}}, (m_l)_{l\in\mathcal{N}}$, $\lim_{l\to\infty}$." Obviously the double sided entropy rate $h_{\infty\times\infty} \leq h_\infty(P)$ and it is easy to construct examples when the inequality is strict. We also introduce a short-hand notation for the KL divergence between the process P and measure $\mathcal{R}$ as,

$$\delta(X_t|\boldsymbol{X}_{<t}) = \mathrm{KL}\left(P(X_t|\boldsymbol{X}_{<t}); \mathcal{R}(X_t|\boldsymbol{X}_{<t})\right).$$

Finally, whenever we use $\mathbb{E}[\delta(X_t|\boldsymbol{X}_{<t})]$, the expectation is taken over $\boldsymbol{X}_{<t}$.

Theorem 1. *For all $m \in \mathcal{N}$ we have,*

$$(i) \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=m}^{T} \frac{\delta(X_{t-m+1}|\boldsymbol{X}_{0:t-m}, \boldsymbol{X}_{t-m+2:t})}{T-m}\right] = 0,$$

$$(ii) \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=m}^{T} \frac{\delta(X_{-t+m-1}|\boldsymbol{X}_{-t:-t+m}, \boldsymbol{X}_{-t+m-2:0})}{T-m}\right] = 0,$$

$$(iii) \lim_{m\to\infty} \lim_{T\to\infty} -\mathbb{E}\left[\sum_{t=m}^{T} \frac{\log \mathcal{R}(X_{t-m+1}|\boldsymbol{X}_{0:t-m}, \boldsymbol{X}_{t-m+2:t})}{T-m}\right] = h_{\infty\times\infty},$$

$$(iv) \lim_{T\to\infty} \lim_{m\to\infty} -\mathbb{E}\left[\sum_{t=m}^{T} \frac{\log \mathcal{R}(X_{t-m+1}|\boldsymbol{X}_{0:t-m}, \boldsymbol{X}_{t-m+2:t})}{T-m}\right] = h_{\infty\times\infty}.$$

The following proof of Theorem 1 relies heavily on the consistency of the $\mathcal{R}$

measure as a predictor [Ryabko, 1988].

Proof. For the first statement, first note that,

$$\begin{aligned}\mathbb{E}\left[\delta(X_{t-m+1}|\boldsymbol{X}_{0:t-m}, \boldsymbol{X}_{t-m+2:t})\right] &= \mathbb{E}\left[\delta(\boldsymbol{X}_{t-m+1:t}|\boldsymbol{X}_{0:t-m}) - \delta(\boldsymbol{X}_{t-m+2:t}|\boldsymbol{X}_{0:t-m})\right] \\ &\leq \mathbb{E}\left[\delta(\boldsymbol{X}_{t-m+1:t}|\boldsymbol{X}_{0:t-m})\right],\end{aligned}$$

where the inequality follows from the fact that the KL divergence is non-negative. The first statement follows from the consistency of $\mathcal{R}$ as a predictor: for every $m \in \mathcal{N}$ we have (see Ryabko [1988, 2008]),

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=m}^{T} \delta(\boldsymbol{X}_{t-m+1:t}|\boldsymbol{X}_{0:t-m})\right] \to 0. \tag{2.9}$$

The proof of the second statement is analogous to that of the first, except that we need the consistency of $\mathcal{R}$ as a predictor "backwards". That is, when the sequence extends to the past rather than to the future. The proof of this consistency is analogous to the proof of the usual (forward) consistency (2.9). Since it is important for exposing some further ideas, we give it here. We consider the case $m = 1$. The general case follows by replacing $Y_i = X_{i:i+m}$ for every i and noting that if the distribution of X_i is stationary–ergodic, then so is the distribution of Y_i. We have,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{T} \delta(X_{-t}|\boldsymbol{X}_{0:-t+1})\right] &= -\sum_{t=0}^{T} \mathbb{E}_{\boldsymbol{X}_{0:-t+1}}\left[\mathbb{E}_{X_{-t}}\left[\log\left(\frac{\mathcal{R}(X_{-t}|\boldsymbol{X}_{0:-t+1})}{P(X_{-t}|\boldsymbol{X}_{0:-t+1})}\right)\right]\right] \\ &= -\mathbb{E}_{\boldsymbol{X}_{0:-T}}\left[\log\left(\frac{\mathcal{R}(\boldsymbol{X}_{-T:0})}{P(\boldsymbol{X}_{-T:0})}\right)\right] \\ &= -\mathbb{E}\left[\log \mathcal{R}(\boldsymbol{X}_{-T:0})\right] + \mathbb{E}\left[\log P(\boldsymbol{X}_{-T:0})\right].\end{aligned}$$

Noting that $\mathbb{E}\left[\sum_{t=0}^{T} \delta(X_{-t}|\boldsymbol{X}_{0:-t+1})\right]$ is non-negative, it is enough to show that $\lim -\frac{1}{T}\mathbb{E}\left[\log \mathcal{R}(\boldsymbol{X}_{-T:0})\right] \leq h_\infty(P)$ to establish the consistency statement.

For every $k \in \mathcal{N}$,

$$\begin{aligned}
-\mathbb{E}\left[\log \mathcal{R}(\boldsymbol{X}_{-T:0})\right] - Th_\infty(P) &= -\mathbb{E}\left[\log \sum_{i=1}^{\infty} w_{i+1}\mathcal{K}^i(\boldsymbol{X}_{-T:0})\right] - Th_\infty(P) \\
&= -\mathbb{E}\left[\log \sum_{i=1}^{\infty} w_{i+1}\mathcal{K}^i(\boldsymbol{X}_{-T:0})\right] - Th_k(P) + Th_k(P) - Th_\infty(P) \\
&\leq -\mathbb{E}\left[\log \mathcal{K}^k(\boldsymbol{X}_{-T:0})\right] - Th_k(P) - \log w_{k+1} + T\varepsilon_k \\
&= o(T) + T\varepsilon_k,
\end{aligned}$$

where $\varepsilon_k = h_k(P) - h_\infty(P)$ and the last equality follows from the consistency of $\mathcal{K}^k$. Since the statement holds for each $k \in \mathcal{N}$, it remains to notice that $\varepsilon_k \to 0$. The third statement follows from the first by noting that,

$$\mathbb{E}\left[\delta(X_{t-m+1}|\boldsymbol{X}_{0:t-m}, \boldsymbol{X}_{t-m+2:t})\right] = -\mathbb{E}\left[\log \mathcal{R}(\boldsymbol{X}_{t-m+1}|\boldsymbol{X}_{0:t-m}, \boldsymbol{X}_{t-m+2:t})\right] + h_{t,m},$$

and that by definition $\lim_{t,m\to\infty} h_{t,m} = h_{\infty\times\infty}$. Analogously, the fourth statement follows from the second, where we additionally need the stationarity of P to shift the time-index by t to the right. □

One could wish for a stronger consistency statement than those established in Theorem 1. For example, a statement that we would like to demonstrate is the following,

$$\lim_{m,t\to\infty} -\mathbb{E}\left[\log \mathcal{R}(X_0|\boldsymbol{X}_{-t:-1}, \boldsymbol{X}_{1:m})\right] = h_{\infty\times\infty}.$$

There are two differences with respect to (*iii*) and (*iv*): first, the limits are taken simultaneously, and, second, there is no averaging over time. We conjecture that this statement is possible to prove. The reason for this conjecture is that it is possible to prove this for some other predictors (other than $\mathcal{R}$). Namely, the consistency proof for the Ornstein predictor [Ornstein, 1978], as well as those of its improvements and generalizations in Morvai et al. [1996] can be extended to our case. These predictors are constructed by taking frequencies of events based on growing segments of the past; however, unlike $\mathcal{R}$, they are very wasteful of data, which is perhaps the reason why they have never been used (to the best

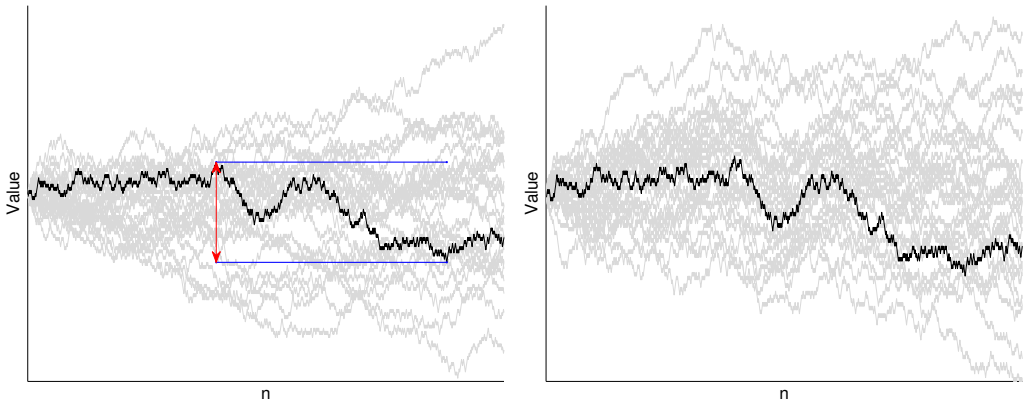

Figure 2.8: (LEFT) Sequences generated from the true process P along with an illustration of the maximum drawdown (RED) on the black sequence. (RIGHT) Bootstrap sequences generated by $\mathcal{R}$-Boot from the black trajectory.

of our knowledge) beyond theoretical analysis. Another possible extension is to prove "almost sure" analogues to the statements of the theorem. This is indeed possible, since the asymptotic consistency holds for $\mathcal{R}$ as a predictor; however, in this case time-averaging is essential, as is also established in Ryabko [1988].

Finally, notice that for standard Bootstrap methods it is often possible to derive asymptotic convergence rates without relatively strong assumptions on the generative process (e.g., exponentially mixing). The class of all stationary–ergodic processes is also so large that this type of analysis is provably impossible. Further, finite-time error bounds (and finite-time optimality analysis) are also impossible since the convergence rates of any non-trivial estimates can be arbitrary slow for this class of processes [Shields, 1996]. Thus, a direct theoretical comparison between $\mathcal{R}$-Boot and other Bootstrap methods using methods which rely on traditional Bootstrap analysis are not possible for this class of processes and we rely on the empirical investigation, in the next section, to evaluate their differences.

5 Empirical Evaluation

An empirical comparison of $\mathcal{R}$-Boot is presented against the circular block Bootstrap[3] and the Markov Bootstrap on both simulated and real datasets with

[3]The circular block Bootstrap has better finite sample properties than the Moving Block Bootstrap because it samples points with equal probability by assuming points lie along a circle,

$B = 1,000$ bootstraps for each method in the estimation of the maximum drawdown statistic, a challenging statistic used in optimization, finance and economics to characterize the drawdown or "adverse excursion" risk. Synthetic sequences are simulated using a real-valued mean-reverting fractional Brownian motion (FBM) process P with mean $\mu = 0$, standard deviation $\sigma = 1$ and Hurst exponent $H = 0.25$ [Mandelbrot and van Ness, 1968][4]. We sample $10,000$ sequences from P of lengths $T_{\text{original}} = 1001$, differencing $(X_t - X_{t-1}, t = 2, \ldots, 1001)$ observations within each sequence into stationary increments of length $T = 1000$. In order to avoid complicated adaptive quantization schemes, which could introduce confounding effects in the result, we use a simple binary discretization: the sequence $\boldsymbol{X}_T$ is such that $X_t = -1$ for negative increments and $X_t = 1$ for positive. From $\boldsymbol{X}_T$, the corresponding *cumulative sequence* $\boldsymbol{Y}_T$, where $Y_t = \sum_{s=1}^{t} X_t$, is computed (representing, e.g., a price sequence). The maximum drawdown is illustrated in Figure 2.8 and defined as,

$$\hat{f}(\boldsymbol{X}_T) = \max_{t=1,\ldots,T} \big(\max_{s=1,\ldots,t} Y_s - Y_t \big), \tag{2.10}$$

and the maximum drawdown $\theta_T = \mathbb{E}\left[\hat{f}(\boldsymbol{X}_T)\right]$ of the process is then computed by averaging the raw estimates $\hat{\theta}_T = \hat{f}(\boldsymbol{X}_T)$ over 10^7 sequences. As θ_T is an increasing function of T, we normalize it by its rate of growth T and compute the estimation error as,

$$\text{MSE}(\mathcal{B}) = \mathbb{E}\left[\frac{(\theta_T - \tilde{\theta}_T^{\mathcal{B}})^2}{T}\right],$$

where the bootstrap estimator is defined as

$$\tilde{\theta}_T^{\mathcal{B}} = \frac{1}{B}\sum_j \hat{f}(\boldsymbol{b}_T^j)$$

and a single bootstrap is defined by $\boldsymbol{b}_T = \mathcal{B}(\boldsymbol{X}_T)$.

---

where blocks extending past T continue along the start of the sample sequence.

[4]The FBM is a parameterized process with stationary increments, long-range dependence and level of self-similarity set using the Hurst index H, where $H = 0.5$ recovers the standard geometric Brownian motion, $H < 0.5$ results in mean-reversion and $H > 0.5$ generates trending behavior.

For circular block Bootstrap, theoretical guidelines provided in the literature (see e.g., Hall et al. [1995], Zvingelis [2003], Politis and White [2004], Patton et al. [2009]) suggest that block width should be of order $O(T^{\frac{1}{3}})$, while for Markov Bootstrap any tuning of the k-order appropriate for the series would require knowledge of the process. In the following, we report results for Markov and circular block Bootstrap methods according to carefully tuned parameters. The block width for each value of T for circular block Bootstrap is optimized in the range of block–widths over $[1, 20]$ (thus always including the theoretical value up to $2n^{\frac{1}{3}}$)[5] and the k-order model size for the Markov Bootstrap is optimized over $[1, 20]$ on all $10,000$ sequences. Notice that such tuning is not possible in practice since only a single sample sequence is available from the true process and the true statistics of the process are obviously unknown. These *best* parameters for circular block Bootstrap and Markov Bootstrap are intended to upper bound the performance that can be achieved by these methods. Furthermore, the best order for Markov Bootstrap also represents an upper bound for any other method using a mixture of Markov models of different order, such as a direct use of the $\mathcal{R}$ measure in Def. 3 to generate sequences sampling from $\mathbb{P}(X_t|\boldsymbol{X}_{<t})$ or the sieve Bootstrap [Bühlmann et al., 1997] that automatically selects the order.

$\mathcal{R}$-Boot[6], circular block Bootstrap and Markov Bootstrap are compared on FBM data in Figure 2.9. circular block Bootstrap is run with its best block width, while Markov Bootstrap is run with its best model size. $\mathcal{R}$-Boot is run with $K_T = \lfloor 1.5 \log(T) \rfloor$ and two values for R, $0.75n$ and $3.5n$. Notice that the largest k-order model used by $\mathcal{R}$-Boot is 4 and always less than the largest model used

[5]The optimal constant in $O(T^{1/3})$ depends on the (unknown) autocovariance and spectral density functions of the process. The automated block width selection procedure in Politis and White [2004], Patton et al. [2009] was tested on a stationary–ergodic process as well as both the easier FBM and currency datasets presented here, but did not perform well. The best block width is not the per realization block length, but the best block width averaged over $10,000$ of the best block widths per specific length in the FBM dataset and separately averaged over the best block width for each of the currency datasets.

[6]No implementations of the measure $\mathcal{R}$ were available, so the heavily optimized extension to the replacement Bootstrap principle for $\mathcal{R}$-Boot in Section 4.4 was implemented in C/C++. Due to the scale of experiments, especially due to the scan of best block–width for the circular block Bootstrap and best k-order Markov model size for the Markov Bootstrap, all algorithms were designed for large scale grid job scheduling through hybrid OpenMP–MPI and deployed on the 9–site Grid 5000 computing infrastructure, www.grid5000.fr. A highly optimized implementation of $\mathcal{R}$-Boot is available in C/C++ upon request.

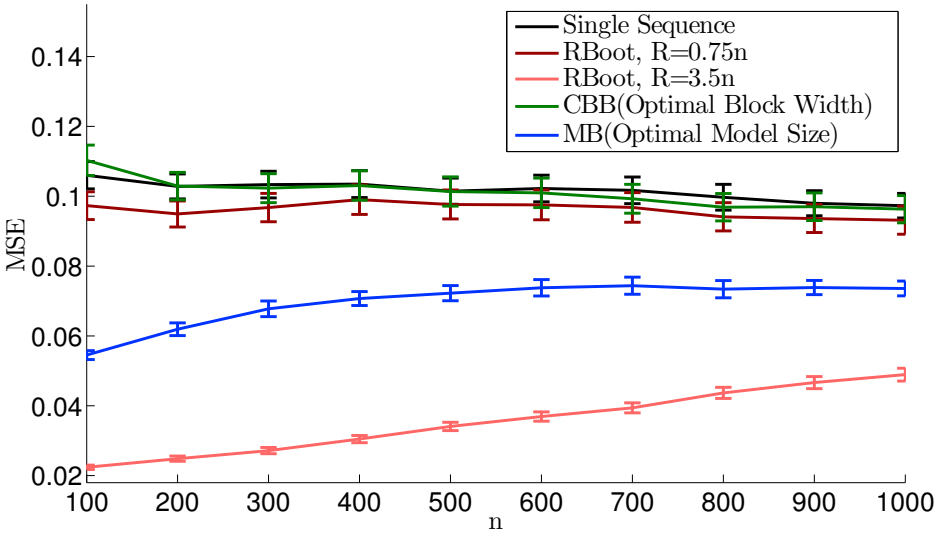

Figure 2.9: MSE on the FBM process for the maximum drawdown statistic.

by the Markov Bootstrap, 20. $\mathcal{R}$-Boot $R = 0.75n$ achieves better performance than circular block Bootstrap and the simple asymptotic estimator $\hat{\theta}_T = \hat{f}(\boldsymbol{X}_T)$ (single sequence), demonstrating that approximated replacement distributions are accurate enough to guarantee bootstraps which resemble the original process. $R = 0.75n$ corresponds with approximately 30% replacements to the original sequence, which generates too little *variability* as compared to Markov Bootstrap, which generates bootstraps from scratch. We increase variability by setting $R = 3.5n$ and notice replacements increase to approximately 140% and $\mathcal{R}$-Boot $R = 3.5n$ significantly outperforms Markov Bootstrap under all values of T. Note that the setting for $R = 3.5n$ was set arbitrarily and further work is necessary to find methods for setting the best R value based on a single observation sequence. As illustrated in the sensitivity analysis that follows, better MSE values are possible for all values of n, given a better setting of R. The increase in MSE with n is indicative of greater variance with the selected value of $R = 3.5n$.

For completeness, we also include the estimation performance of the mean and standard deviation in Figure 2.10, the most common statistics measured using the Bootstrap. These results further demonstrate the superior performance of $\mathcal{R}$-Boot in statistical estimation. While all methods converge to the same level of performance as T increases, $\mathcal{R}$-Boot $R = 3.5n$ replicates the process with enough

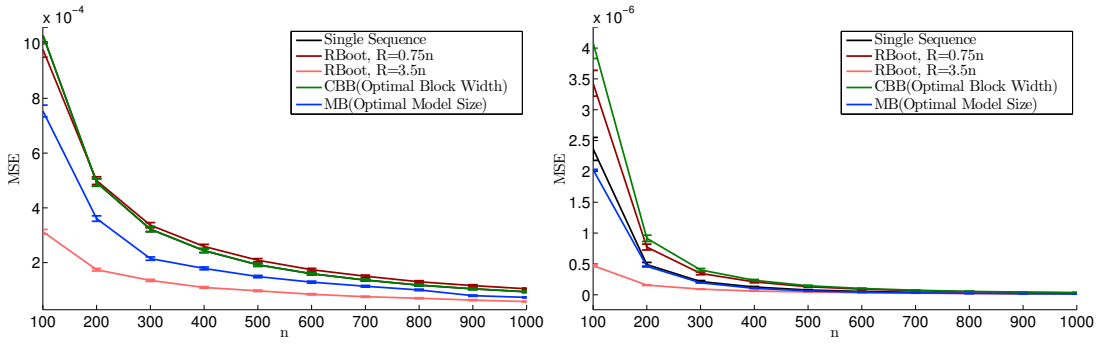

Figure 2.10: Bootstrap estimation performance for the mean (LEFT) and standard deviation(RIGHT).

accuracy to outperform the best performance of other Bootstrap methods by a factor of approximately 2 in the case of the mean and 4 in the case of the standard deviation for very short sequences of length $T = 100$.

A sensitivity analysis of the presented algorithms demonstrates that while sub-optimal tuning negatively impacts both circular block Bootstrap and Markov Bootstrap performance, $\mathcal{R}$-Boot is quite robust in that a wide range of R values consistently outperform circular block Bootstrap and Markov Bootstrap by a significant margin. In the results, we considered two values for parameter R to show $\mathcal{R}$-Boot is competitive w.r.t. careful tuning of circular block Bootstrap and Markov Bootstrap. Here we explore the parameter sensitivity of these three methods, showing the potential *significant* advantage of $\mathcal{R}$-Boot. In Figure 2.11 we report the MSE performance of each method for a full range of parameters. Circular block Bootstrap obtains a very poor performance for block sizes that are too small while an increasing block width improves performance, but as noticed in Section 5, fails to achieve decisively improved performance against the single sequence estimator. On the other hand, Markov Bootstrap significantly outperforms circular block Bootstrap and the single sequence estimator. Nonetheless, Figure 2.11 illustrates the dependence on correct model size specification to good performance. In fact, small orders introduce too much bias (i.e., the process cannot be described accurately enough with 1 or 2-Markov models), while large orders suffer from large variance, where model sizes above the optimal order overfit noise. Furthermore, we notice that the best model size changes with T, where longer se-

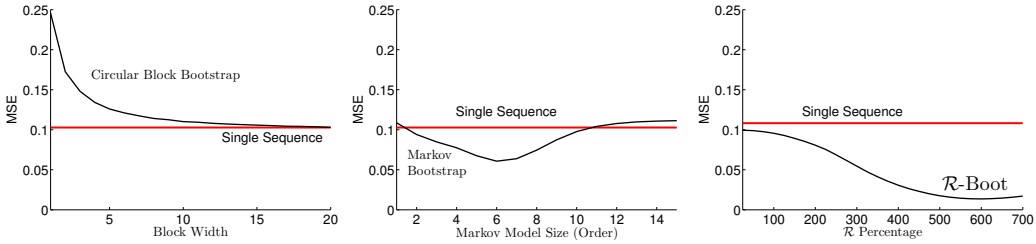

Figure 2.11: Sensitivity analysis of circular block Bootstrap, Markov Bootstrap, and $\mathcal{R}$-Boot with respect to their parameters (block width, Markov order, and number of replacements R as a percentage of T) in the FBM experiment for $T = 200$ (notice the difference in scale for circular block Bootstrap).

quences allow for larger orders without significantly increasing the variance. As a result, properly tuning optimal order Markov Bootstrap from one sequence is challenging and a poor parameter choice can significantly impact performance.

Finally, we report the performance of $\mathcal{R}$-Boot w.r.t. the number of replacements. As discussed in Sect. 4, R corresponds to the number of *attempted* replacements. The need for large values is due to $\mathcal{R}$-Boot's sequential nature. In fact, as the sequence $\boldsymbol{z}_T^r$ changes, the replacements in a specific position t may have different outcomes because of the conditioning in computing $\mathcal{R}_{\boldsymbol{X}_T}(\cdot|\boldsymbol{z}_{<t_r}^{r-1}, \boldsymbol{z}_{>t_r}^{r-1})$. As a result, we need R to be large enough to allow for a sufficient number of actual changes in the original sequence to generate a consistent Bootstrap sequence. In order to provide an intuition about the *actual* replacements, let R' be the number of times the value $z_{t_r}^{r-1}$ is changed across R iterations. For $R = 0.75n$, we obtain on average $R' \approx 0.30n$, meaning that less than 30% of the original sequence is actually changed. Similar results are observed from different values of R in both FBM and the real datasets. As illustrated in Figure 2.11, both choices of R used in the experiments are suboptimal, since the performance further improves for larger R (before deteriorating as the choice of R grows too large). Nonetheless, the change in performance is quite smooth and $\mathcal{R}$-Boot outperforms both optimal block width circular block Bootstrap and optimal model size Markov Bootstrap for a large range of R values.

ESTIMATOR	USD/CHF	EUR/USD	GBP/USD	USD/JPY
Asymptotic (single sequence)	93.0018	66.6727	72.8849	119.1314
Circular Block Bootstrap (optimal)	93.2946 ± 2.0745	66.9138 ± 1.7844	73.1951 ± 3.3746	119.2367 ± 3.319
$\mathcal{R}$-Boot (100% actual replacements)	44.6137 ± 2.582	31.7459 ± 1.661	37.0970 ± 2.2582	50.5639 ± 2.5812
Markov Bootstrap (optimal)	43.7268 ± 2.3188	29.4964 ± 1.537	36.5849 ± 2.126	47.5460 ± 2.3711

Figure 2.12: Maximum drawdown MSE performance (with standard errors) on multiple real datasets.

5.1 Currency Datasets

We proceed to test $\mathcal{R}$-Boot on real data; namely, differenced high-frequency 1-minute currency pair data. Currency pairs are relative value comparisons between two currencies. We assume the differenced, and therefore stationarized, series is ergodic according to Bassler et al. [2007]. This data is useful for analysis because of its availability, high liquidity, 24-hour nature and minimal gaps in the data due to non-trading times. This final feature is important because it reduces the noise caused by activities during non-trading hours, such as stock or economic news. Although these series do not strictly conform to the stationary–ergodic assumptions needed for $\mathcal{R}$-Boot to work well, we evaluate these approaches and report the results.

We estimate the maximum drawdown statistic using Bootstrap methods on four pairs of currencies considering the ratio of the first currency over the second currency. For example, the British Pound to U.S. Dollar cross is calculated as GBP/USD. We work with the one-minute closing price. One-minute data is a compressed representation of the actual sequence which includes four data points for each one-minute time interval: the open, high, low and close prices. The data[7] used in this chapter was segmented into single day blocks of $T = 1440$ minutes. Days which were partially open due to holidays or announced closures were removed from the data set for consistency. A total of 300 days (1.5 years excluding holidays and weekends) of daily sequential samples from the underlying daily generative process were used for each currency pair.

In Figure 2.12, we report estimation performance across all the currency datasets for best order Markov Bootstrap (with model size selected in $[1, 20]$),

[7] We downloaded the one-minute historical data from http://www.forexhistorydatabase.com/.

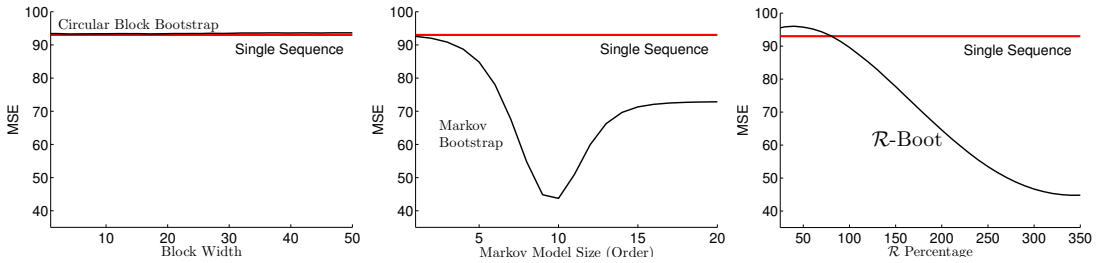

Figure 2.13: Sensitivity to parameters for different Bootstrap methods for the USDCHF currency (notice the difference in scale for circular block Bootstrap).

best block width circular block Bootstrap (with width selected in $[1, 20]$), and $\mathcal{R}$-Boot with $R = 3.5n$ (chosen to match the value used in the FBM experiments). As we noted in the FBM analysis, a large value of R does not necessarily correspond to a large value of replacements. In fact, we observed that the actual number of replacements in the FBM sequences with $R = 3.5n$ were approximately 140% replacements. On these datasets, circular block Bootstrap has constant behavior and cannot beat the simple asymptotic estimator (single sequence). On the other hand, $\mathcal{R}$-Boot and Markov Bootstrap significantly improve the maximum drawdown estimation and they always perform similarly across different currencies (notice that the small advantage of Markov Bootstrap is never statistically significant).

The full range of parameters for each of these methods is reported in Figure 2.13. These results mostly confirm the analysis on synthetic data, where Markov Bootstrap achieves very good performance for a specific range of model sizes, while performance rapidly degrades in model sizes that are too large and too small. Finally, $\mathcal{R}$-Boot confirms a similar behavior, as in the FBM, with a performance which gracefully improves as R increases with a gradual degradation thereafter.

6 Conclusions

Statistical estimation on a single short dependent time series sequence is hard. The Bootstrap is one of few tools capable of handling this estimation problem.

Consistency results for existing methods rely on restrictive assumptions on the generative process. Convergence rates depend on optimally set parameters. These methods are sensitive to correct parameter specification. Poor settings result in poor performance. This was demonstrated in the sensitivity analysis in Section 5 and 5.1. Further, we demonstrated that even when parameters are optimally set, block methods completely fail as estimators of complex statistics. It is also clear that even when over-fitting for the optimal Markov model size per value T, the Markov Bootstrap is sensitive to correct model specification. It is clear that existing methods do not perform well for dependent processes, complex statistics or short sequences that do not reveal the full structure of the process. We approached this problem by introducing $\mathcal{R}$-Boot, an iterative replacement Bootstrap algorithm. $\mathcal{R}$-Boot successfully managed these challenging circumstances under several synthetic and real world datasets. The basis of $\mathcal{R}$-Boot is the novel *replacement Bootstrap principle* presented in this chapter. This principle generates bootstrap sequences by simultaneously replacing symbols using an estimated replacement distribution. Preliminary theoretical and empirical results suggest that the replacement Bootstrap can significantly improve the estimation of complicated statistics in the general class of stationary–ergodic processes.

7 Future Work

$\mathcal{R}$-Boot incrementally approximates the simultaneous replacements in the replacement Bootstrap. Empirical results in Section 5 are promising. An intermediate approach between incremental and simultaneous replacements can be achieved using blocks. The sample can be mapped from a single symbol sequence to a sequence of blocks. $\mathcal{R}$-Boot can then be run on the new symbols to achieve block replacements. Another extension is to construct mixtures over multiple block sizes. We leave this for future work.

The measure $\mathcal{R}$ is designed for stationary–ergodic processes. No known rates exist for processes with specific structure. We conjecture that other replacement Bootstrap algorithms are possible using alternative density estimation methods.

Future work should also extend theoretical guarantees. Specifically, guarantees on the Bootstrap process and processes with specific structure (e.g., mixing, auto-regressive). Future work on computational complexity should focus on reducing redundant calculations across bootstraps. This can significantly reduce computations by finding and removing overlapping calculations. Finally, another extension might consider extending the measure $\mathcal{R}$ over several time frames.

The conditional replacement distribution estimation step in $\mathcal{R}$-Boot is equivalent to model-free distribution-based missing data *imputation* [Van Buuren, 2012]. It is natural to study this relationship and extend $\mathcal{R}$-Boot to this problem. Equivalently, it seems reasonable to leverage the estimated conditional replacement distributions in *Outlier Detection*. Each replacement step results in a conditional distribution over symbols and reveals insights on the probability of events in a time series.

An alternative application is to Bootstrap ensemble outputs. Ensemble performance is often limited by limited sample size. Bootstrapping ensemble outputs would approximate the dependent output sequences. These bootstrap sequences would be very useful in "Model Compression" Bucilua et al. [2006]. The performance of "Model compression" depends directly on sample size. A common practice is to generate pseudo samples through convolutions or noise. This is problematic as it does not approximate the true process. Bootstrap samples avoid this problem by generated samples from the sampling distribution.

Chapter 3

Risk Averse Multi-Arm Bandits

Contents

1 Introduction 49
2 The Multi–Arm Bandit Problem 53
3 Mean–Variance Multi–arm Bandit 57
4 Mean–Variance Lower Confidence Bound Algorithm . . . 65
5 Exploration–Exploitation Algorithm 72
6 Numerical Simulations 77
7 Sensitivity Analysis 78
8 Discussion 82
9 Conclusions 83
10 Subsequent Work 85

1 Introduction

In the previous chapter, we introduced the novel $\mathcal{R}$-Boot iterative Bootstrap algorithm to address this problem for the most general class of stationary–ergodic processes and demonstrated its performance on the particularly challenging maximum drawdown statistic. Though we studied the problem of estimation in the previous chapter, we did not study the performance of interleaving the estimation of a risk objective and decision-making. Here, we study the multi-arm bandit problem [Robbins and Monro, 1951, Thompson, 1933, 1935], which naturally characterizes this problem as the exploration-exploitation dilemma. By only evaluating

the risk-averse policy according to observations it selects, the uncertainty, resulting from partial observation of the choice distributions, can be directly attributed to the performance of the policy. Additionally, we revert to a simple risk-averse objective (Markowitz [1952] Mean-Variance) for which estimation is unbiased and efficient, to avoid problems with statistical estimation. Thus, since statistical estimation is no longer a problem, we can directly study the policy performance under uncertainty.

The multi–armed bandit [Robbins, 1952] elegantly formalizes the problem of online learning with partial feedback, which encompasses a large number of real–world applications, such as clinical trials [Robbins and Monro, 1951], online advertising [Amin et al., 2012], adaptive routing [Avner et al., 2012], cognitive radio [Gai et al., 2010] and auction mechanisms [Gonen and Pavlov, 2007]. In this setting, a learner chooses among several arms (e.g., different treatments), each characterized by an independent reward distribution (e.g., the treatment effectiveness). At each point in time, the learner selects one arm and receives a noisy reward observation from that arm (e.g., the effect of the treatment on one patient). This is the partial information nature of the setting. This process repeats until a known *fixed* horizon or unknown *anytime* horizon. Given a finite horizon (e.g., number of patients involved in the clinical trial), the learner faces a dilemma between repeatedly exploring all arms (treatments) to collect reward information versus exploiting current reward estimates by selecting the arm with the highest estimated reward (most effective treatment observed so far). The standard objective (expectation maximization) relies on unbiased estimates of the mean, so it ignores the estimation problem in Chapter 2. The learning objective is to solve this exploration–exploitation dilemma by simultaneously accumulating as much reward as possible, while minimizing cumulative *regret*. Regret accumulates from having pulled suboptimal arms, where the per-step regret is defined as the difference between the selected arms and the optimal arm in hindsight. A positive result is defined by an algorithm that has a per-step regret that goes to zero as time grows. This algorithm is then referred to as a "no-regret" algorithm.

Many algorithms have been developed around this simple objective. In par-

ticular, Thompson sampling [Chapelle and Li, 2011] and upper confidence bound (UCB) [Auer et al., 2002] algorithms have been shown to have logarithmic regret, which is known to be optimal [Lai and Robbins, 1985]. UCB algorithms use the "Optimism in the Face of Uncertainty" principle introduced by Lai and Robbins [1985] to select arms based on their UCB. Thompson sampling assumes a prior distribution over arms and uses *probability matching* to select arms. After each realization, a conditional probability distribution of the mean is updated for the selected arm. Arm observations are then simulated from the estimated distribution and the arm with the highest simulated mean is selected. This chapter focuses on algorithms using a UCB strategy[1].

In many practical problems, maximizing the expectation may not be the most desirable objective. Solutions that guarantee high rewards in expectation may be too "risky" from a risk-averse perspective. For instance, in clinical trials, the treatment which works best *on average* might also have considerable *variability*; resulting in adverse side effects for some patients. In the standard objective, treatments with equal means and contrasting variance are treated equally. Risk-aversion weigh these arms according to some measure of risk. If variance measures an arm's risk, less variance equates to less risk. In this particular example, a treatment which is less effective on average, but consistently effective on different patients, may be preferable w.r.t. an effective but risky treatment. This chapter introduces the Markowitz [1952] Mean–Variance objective to the stochastic multi-arm bandit setting with the aim of studying the impact of risk–averse objectives on the exploration–exploitation dilemma. Recall that the choice of working with the mean and variance is driven by a desire to study the influence of online estimation on decision-making. The mean and variance are both unbiased estimators and allows us to avoid problems with estimation.

This work is the first to study online estimation of a risk objective in the stochastic multi-arm bandit problem. The only other analysis studies estimation risk, which is unrelated to our study. In particular, Audibert et al. [2009] analyze

[1]For a survey of the multi-arm bandit, please see e.g., Cesa-Bianchi and Lugosi [2003]. For a review of UCB algorithms, please see e.g., Bubeck and Cesa-Bianchi [2012]. For a review of Thompson sampling, please see e.g., Kaufmann et al. [2012]

the distribution of the pseudo-regret when the regret deviates from its expectation, revealing that an anytime version of UCB algorithms based only on the sample mean *UCB1*[Auer et al., 2002], and an extension based on empirical Bernstein bounds relying also on the sample variance *UCB-V* [Audibert et al., 2009], might have large regret with some non-negligible probability[2]. They note that an anytime horizon suffers a greater risk of deviation due to an uncertainty associated to the evaluation time, while a fixed horizon effectively manages this deviation risk because the evaluation time is known from the first round. Without a clearly defined evaluation time, the anytime setting challenges how the exploration–exploitation should be managed. Ultimately, better concentration around the expectation can be achieved by adapting the "aggressiveness" of the exploration rate to reduce the risk of deviating from the expected performance. This result applies generally to any objective relying on empirical estimates and a UCB and not to specifically to the expectation–maximization objective being studied. Salomon and Audibert [2011] extended this analysis to prove negative results showing that no *anytime* algorithm can achieve a regret with both a small expected regret and exponential tails (i.e., low regret in high probability).

In Section 2, we introduce notation and the stochastic multi-arm bandit problem, reviewing existing results in the (standard) expectation maximization objective. Section 3 introduces additional notation and define the Mean–Variance bandit problem, where we introduce a novel risk-averse objective. In Section 4 we introduce a confidence–bound algorithm for this risk-averse objective and study its theoretical properties. In Section 5 we introduce a (non-UCB) algorithm that explicitly splits exploration and exploitation phases. Numerical results on synthetic data validating the theoretical results are reported in Section 6. Section 7 presents a sensitivity analysis, Section 8 provides a brief discussion, and finally, Section 9 concludes the chapter with suggestions on future work. We briefly review their contribution to our work in Section 10.

[2]Although the analysis is mostly directed to the pseudo–regret, as commented in Remark 2 at page 23 of Audibert et al. [2009], it can be extended to the true regret.

2 The Multi–Arm Bandit Problem

In the following, we present notation and review the relevant literature on the stochastic multi-arm bandit problem and risk in more detail.

2.1 Notation, Setting and Definitions

The fixed horizon stochastic multi–arm bandit problem is defined over T rounds and considers K independent arms, each characterized by a distribution ν_i, with mean μ_i and variance σ_i^2, with observations (rewards) X_i bounded in the interval $[0, b]$. We denote by $X_{i,s} \sim \nu_i$ the s-th i.i.d. random reward observation drawn from the distribution of arm i. At each round t, an algorithm selects arm I_t and observes sample $Z_t = X_{I_t,N_{i,t}}$, where $N_{i,t}$ is the number of samples observed from arm i up to time t (i.e., $N_{i,t} = \sum_{s=1}^{t} \mathbb{I}\{I_t = i\}$) and the aim is to select the *optimal* arm i^* having the largest *expected* reward

$$\mu_{i^*} = \max_{i=1,\dots,K} \mu_i.$$

Given $\{X_{i,s}\}_{s=1}^{t}$ i.i.d. samples from the distribution ν_i, we define the empirical mean of an arm i with t samples as,

$$\hat{\mu}_{i,t} = \frac{1}{t} \sum_{s=1}^{t} X_{i,s},$$

and the empirical variance as,

$$\hat{\sigma}_{i,t}^2 = \frac{1}{t} \sum_{s=1}^{t} \big(X_{i,s} - \hat{\mu}_{i,t}\big)^2. \tag{3.1}$$

The empirical mean for learning algorithm $\mathcal{A}$ over T rounds is defined as,

$$\hat{\mu}_T(\mathcal{A}) = \frac{1}{T} \sum_{t=1}^{T} Z_t. \tag{3.2}$$

We measure the performance of the learning algorithm $\mathcal{A}$ according to its cumulative regret,

$$\mathcal{R}_T(\mathcal{A}) = \sum_{t=1}^{T} X_{i^*,t} - \sum_{t=1}^{T} Z_t. \tag{3.3}$$

The aim is for a policy to have an expected (cumulative) regret $\mathbb{E}[\mathcal{R}_T]$ that is as small as possible, which is equivalent to maximizing the total expected reward achieved up to time T. Accordingly, the expected regret can be expressed as,

$$\mathbb{E}[\mathcal{R}_T(\mathcal{A})] \triangleq \sum_{i=1}^{K} \mathbb{E}[N_{i,T}]\Delta_i, \tag{3.4}$$

where $\Delta_i = \mu_{i^*,T} - \mu_T(\mathcal{A})$ is the expected loss of playing arm i. Hence, a policy that aims at minimizing the expected regret should minimize the expected sampling times of suboptimal arms over the horizon.

2.2 Optimism in the Face of Uncertainty Principle

Lai and Robbins [1985] introduced parametric UCB algorithms within a minimax framework as an approach to solving the stochastic multi–arm bandit problem. This approach follows what is referred to as the "optimism in the face of uncertainty principle". These algorithms use sample means along with a UCB on each arm which concentrate according to the number of samples drawn from each arm. As long as an arm is never chosen, its bound is infinite, so the aim of UCB algorithms is to explore the possible arms with the aim of identifying the arm with the highest expected reward as fast as possible. The algorithm exploits the arm with the highest expectation, with some probability. It is also natural to *initialize* arm estimates with a minimum number of samples. This is an unavoidable cost in this setting. The cumulative regret of UCB algorithms grows with order $\mathcal{O}(\log T)$. UCB algorithms use bounds based on the empirical mean, as in the *UCB1* algorithm [Auer et al., 2002]. The pseudocode for *UCB1* is presented in Figure 3.1.

The expected regret is presented in Theorem 2.

Input:

- Rounds T
- Arms $1, \ldots, K$

For all $t = 1, \ldots, T$**, repeat**

1. **For all** $i = 1, \ldots, K$**, repeat**

$$B_{i,N_{i,t-1}} = \hat{\mu}_{i,N_{i,t-1}} + b\sqrt{\frac{\log T}{N_{i,t-1}}}$$

   end for
2. Learner chooses $I_t = \arg\max_{i=1,\ldots,K} B_{i,N_{i,t-1}}$
3. Learner updates $N_{I_t,t} = N_{I_t,t-1} + 1$
4. Learner observes $X_{I_t,N_{I_t,t}} \sim \nu_{I_t}$
5. Learner updates $\hat{\mu}_{I_t,N_{I_t,t}}$

end for

Figure 3.1: *UCB1*

Theorem 2. *The expected pseudo-regret for UCB1 [Auer et al., 2002] defined by the upper confidence bound,*

$$B_{i,N_{i,t-1}} = \hat{\mu}_{i,N_{i,t-1}} + b\sqrt{\frac{\log T}{N_{i,t-1}}},$$

satisfies,

$$\mathbb{E}[\mathcal{R}_T] \leq \sum_{i:\Delta_i>0} \left[\frac{8b^2}{\Delta_i}\log T + 2\Delta_i\right].$$

Audibert et al. [2009] provide a thorough analysis of UCB algorithms, while introducing empirical Bernstein bounds based on the empirical variance. Unlike *UCB1*, the index policy of *UCB-V* considers both the empirical mean $\hat{\mu}_{i,t}$ and empirical variance,

$$\hat{\sigma}^2_{i,t} = \frac{1}{t}\sum_{s=1}^{t}(X_{i,s} - \hat{\mu}_{i,t})^2.$$

An explicit regret bound is presented in Theorem 3, while pseudo-code is presented in Figure 3.2. Audibert et al. [2009] show that algorithms using the empirical

Input:

- Rounds T
- Arms $1, \ldots, K$

For all $t = 1, \ldots, T$**, repeat**

1. **For all** $i = 1, \ldots, K$**, repeat**

$$B_{i,N_{i,t-1}} = \hat{\mu}_{i,N_{i,t-1}} + \sqrt{\frac{2\zeta\hat{\sigma}^2_{k,N_{i,t-1}} \log T}{N_{i,t-1}}} + c\frac{3\zeta \log T}{N_{i,t-1}}$$

   end for
2. Learner chooses $I_t = \arg\max_{i=1,\ldots,K} B_{i,N_{i,t-1}}$
3. Learner updates $N_{I_t,t} = N_{I_t,t-1} + 1$
4. Learner observes $X_{I_t,N_{I_t,t}} \sim \nu_{I_t}$
5. Learner updates $\hat{\mu}_{I_t,N_{I_t,t}}$ and $\hat{\sigma}^2_{I_t,N_{I_t,t}}$

end for

Figure 3.2: *UCB-V*

variance outperform those that only rely on the empirical mean, as long as the variance of suboptimal arms is much smaller than the squared upper bound on rewards, b^2.

Theorem 3. *Let $c = 1$ and $\varepsilon = \zeta \log t$, for $\zeta > 1$. Then there exists a constant c_ζ that depends on ζ only such that for any $K \geq 2$, the expected pseudo-regret for UCB–V [Audibert et al., 2009] defined by the upper confidence bound,*

$$B_{i,N_{i,t-1}} = \hat{\mu}_{i,N_{i,t-1}} + \sqrt{\frac{2\zeta\hat{\sigma}^2_{i,N_{i,t-1}} \log T}{N_{i,t-1}}} + c\frac{3\zeta \log T}{N_{i,t-1}},$$

satisfies

$$\mathbb{E}[\mathcal{R}_T] \leq c_\zeta \sum_{i:\Delta_i>0} \left(\frac{\sigma_i^2}{\Delta_i^2} + 2\right) \log T. \tag{3.5}$$

For instance, for $\zeta = 1.2$, the result holds with $c_\zeta = 10$.

3 Mean–Variance Multi–arm Bandit

3.1 Additional Notation, Setting and Definitions

The Mean–Variance multi–arm bandit problem is defined similarly to the standard stochastic multi-arm bandit. While in the standard objective, the aim is to select the arm leading to the highest reward in *expectation* (the arm with the largest expected value μ_i), here we focus on the problem of finding the arm that efficiently trades off risk versus reward (risk–reward). Although many risk objectives have been proposed, here we focus on the Mean–Variance model proposed by Markowitz [1952], where the empirical means represent the arm reward value and empirical variance represents the arm risk.

Definition 4. *The Mean–Variance of an arm i with mean μ_i, variance σ_i^2 and coefficient of absolute risk tolerance ρ is defined as*[3] $\mathrm{MV}_i = \sigma_i^2 - \rho\mu_i$.

Thus, it easily follows that the arm best minimizing the Mean–Variance is,

$$i^* = \arg\min_{i=1,\dots,K} \mathrm{MV}_i.$$

We notice that we can obtain a full range of settings according to the value of risk tolerance ρ. As $\rho \to \infty$, the Mean–Variance of arm i tends to the opposite of its expected value μ_i, with the objective reducing to the standard expected reward maximization setting traditionally considered in multi–arm bandit problems. With $\rho = 0$, the Mean–Variance reduces to minimizing the variance σ_i^2, where the objective becomes variance minimization.

Given $\{X_{i,s}\}_{s=1}^{t}$ i.i.d. samples from the distribution ν_i, we define the empirical Mean–Variance of an arm i with t samples as,

$$\widehat{\mathrm{MV}}_{i,t} = \hat{\sigma}_{i,t}^2 - \rho\hat{\mu}_{i,t},$$

We now consider a learning algorithm $\mathcal{A}$ and its corresponding performance over

[3] The coefficient of risk tolerance is the inverse of the Pratt [1964] coefficient of absolute risk aversion $A = \frac{1}{\rho}$.

T rounds. Similar to a single arm i, we define its empirical Mean–Variance as,

$$\widehat{\mathrm{MV}}_T(\mathcal{A}) = \hat{\sigma}_T^2(\mathcal{A}) - \rho\hat{\mu}_T(\mathcal{A}), \tag{3.6}$$

where the variance of an algorithm is defined as,

$$\hat{\sigma}_T^2(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T}\big(Z_t - \hat{\mu}_T(\mathcal{A})\big)^2. \tag{3.7}$$

This leads to a natural definition of the (random) regret at each single run of the algorithm as the difference in the Mean–Variance performance of the algorithm compared to the best arm.

Definition 5. *The regret for a learning algorithm $\mathcal{A}$ over T rounds is defined as,*

$$\mathcal{R}_T(\mathcal{A}) = \widehat{\mathrm{MV}}_T(\mathcal{A}) - \widehat{\mathrm{MV}}_{i^*,T}. \tag{3.8}$$

Given this definition, the objective is to design an algorithm whose regret decreases as the number of rounds increases (in high probability or in expectation). We notice that the previous definition actually depends on *unobserved* samples. In fact, $\widehat{\mathrm{MV}}_{i^*,T}$ is computed on T samples i^* which are not actually observed when running $\mathcal{A}$. This matches the definition of *true* regret in standard bandits (see e.g., Audibert et al. [2009]). Thus, in order to clarify the main components characterizing the regret, we introduce additional notation. Let,

$$Y_{i,t} = \begin{cases} X_{i^*,t} & \text{if } i = i^* \\ X_{i^*,t'} \text{ with } t' = N_{i^*,T} + \sum\limits_{j\neq i, j\neq i^*} N_{j,T} + t, & \text{otherwise} \end{cases}$$

be a renaming of the samples from the optimal arm, such that while the algorithm was pulling arm i for the t-th time, $Y_{i,t}$ is the *unobserved sample from* i^*. Then we define the corresponding mean and variance as,

$$\tilde{\mu}_{i,N_{i,T}} = \frac{1}{N_{i,T}}\sum_{t=1}^{N_{i,T}} Y_{i,t}, \qquad \tilde{\sigma}^2_{i,N_{i,T}} = \frac{1}{N_{i,T}}\sum_{t=1}^{N_{i,T}}\big(Y_{i,t} - \tilde{\mu}_{i,N_{i,T}}\big)^2. \tag{3.9}$$

Given these additional definitions, we can now rewrite the regret as,

$$\mathcal{R}_T(\mathcal{A}) = \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\left[(\hat{\sigma}^2_{i,N_{i,T}} - \rho\hat{\mu}_{i,N_{i,T}}) - (\tilde{\sigma}^2_{i,N_{i,T}} - \rho\tilde{\mu}_{i,N_{i,T}})\right] + \frac{1}{T}\sum_{i=1}^{K} N_{i,T}\big(\hat{\mu}_{i,N_{i,T}} - \hat{\mu}_T(\mathcal{A})\big)^2 - \frac{1}{T}\sum_{i=1}^{K} N_{i,T}\big(\tilde{\mu}_{i,N_{i,T}} - \hat{\mu}_{i^*,T}\big)^2. \quad (3.10)$$

Since the last term is always negative and small[4], our analysis focuses on the first two terms which reveal two interesting characteristics of $\mathcal{A}$. First, an algorithm $\mathcal{A}$ suffers a regret whenever it chooses a suboptimal arm $i \neq i^*$ and the regret corresponds to the difference in the empirical Mean–Variance of i w.r.t. the optimal arm i^*. Such a definition has a strong similarity with the definition of regret in 3.3, where i^* is the arm with the highest expected value and the regret depends on the number of times suboptimal arms are pulled and their respective gaps w.r.t. the optimal arm i^*. In contrast to the standard formulation of regret, $\mathcal{A}$ also suffers an additional regret from the variance $\hat{\sigma}^2_T(\mathcal{A})$, which depends on the variability of pulls $N_{i,T}$ over different arms. Recalling the definition of the mean $\hat{\mu}_T(\mathcal{A})$ as the weighted mean of the empirical means $\hat{\mu}_{i,N_{i,T}}$ with weights $\frac{N_{i,T}}{T}$ (see eq. 3.7), we notice that this second term is a weighted variance of the means and represents a penalty associated with the algorithm switching between arms between rounds. In fact, if an algorithm simply selects and pulls a single arm from the beginning, it would not suffer any penalty from this term (secondary regret), since $\hat{\mu}_T(\mathcal{A})$ would coincide with $\hat{\mu}_{i,N_{i,T}}$ for the chosen arm and all other components would have zero weight. On the other hand, an algorithm accumulates this "switching" cost as the mean $\hat{\mu}_T(\mathcal{A})$ deviates from any specific arm; where the maximum penalty peaks at the mean $\hat{\mu}_T(\mathcal{A})$ furthest from all arm means. This makes sense in that it suggests an algorithm that equally pulls all arms has no preference for any of the arms, so it fails to identify any of the arms as optimal. In the next sections we introduce and study two simple algorithms. We study how well they trade-off the two components of the regret.

[4]More precisely, it can be shown that this term decreases with rate $\mathcal{O}\left(\frac{K\log(\frac{1}{\delta})}{T}\right)$ with probability $1-\delta$.

The previous definition of regret can be further elaborated to obtain the upper bound

$$\mathcal{R}_T(\mathcal{A}) \leq \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\widehat{\Delta}_i + \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\widehat{\Gamma}^2_{i,j}, \tag{3.11}$$

where

$$\widehat{\Delta}_i = (\hat{\sigma}^2_{i,N_{i,T}} - \tilde{\sigma}^2_{i,N_{i,T}}) - \rho(\hat{\mu}_{i,N_{i,T}} - \tilde{\mu}_{i,N_{i,T}}),$$

and

$$\widehat{\Gamma}^2_{i,j} = (\hat{\mu}_{i,N_{i,T}} - \hat{\mu}_{j,N_{j,T}})^2,$$

First, we elaborate on the two mean terms in the regret as,

$$\begin{aligned}\hat{\mu}_{i^*,T} &= \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}} Y_{i,t} \\ &= \frac{1}{T}\sum_{i=1}^{K} N_{i,T}\tilde{\mu}_{i,N_{i,T}},\end{aligned}$$

and

$$\begin{aligned}\hat{\mu}_T(\mathcal{A}) &= \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}} X_{i,t} \\ &= \frac{1}{T}\sum_{i=1}^{K} N_{i,T}\hat{\mu}_{i,N_{i,T}}.\end{aligned}$$

Similarly, the two variance terms can be written as,

$$\begin{aligned}\hat{\sigma}^2_T(\mathcal{A}) &= \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}} \big(X_{i,t} - \hat{\mu}_T(\mathcal{A})\big)^2 \\ &= \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}} \big(X_{i,t} - \hat{\mu}_{i,N_{i,T}}\big)^2 \\ &+ \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}} \big(\hat{\mu}_{i,N_{i,T}} - \hat{\mu}_T(\mathcal{A})\big)^2 + \frac{2}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}} \big(X_{i,t} - \hat{\mu}_{i,N_{i,T}}\big)\big(\hat{\mu}_{i,N_{i,T}} - \hat{\mu}_T(\mathcal{A})\big) \\ &= \frac{1}{T}\sum_{i=1}^{K} N_{i,T}\hat{\sigma}^2_{i,N_{i,T}} + \frac{1}{T}\sum_{i=1}^{K} N_{i,T}\big(\hat{\mu}_{i,N_{i,T}} - \hat{\mu}_T(\mathcal{A})\big)^2 + 0,\end{aligned}$$

and

$$\begin{aligned}\sigma^2_{i^*,T} &= \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}}\big(Y_{i,t}-\hat{\mu}_{i^*,T}\big)^2 \\ &= \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}}\big(Y_{i,t}-\tilde{\mu}_{i,N_{i,T}}\big)^2 \\ &+ \frac{1}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}}\big(\tilde{\mu}_{i,N_{i,T}}-\hat{\mu}_{i^*,T}\big)^2 + \frac{2}{T}\sum_{i=1}^{K}\sum_{t=1}^{N_{i,T}}\big(Y_{i,t}-\tilde{\mu}_{i,N_{i,T}}\big)\big(\tilde{\mu}_{i,N_{i,T}}-\hat{\mu}_{i^*,T}\big) \\ &= \frac{1}{T}\sum_{i=1}^{K}N_{i,t}\tilde{\sigma}^2_{i,N_{i,T}} + \frac{1}{T}\sum_{i=1}^{K}N_{i,T}\big(\tilde{\mu}_{i,N_{i,T}}-\hat{\mu}_{i^*,T}\big)^2 + 0.\end{aligned}$$

Combining these terms, we obtain the following regret,

$$\begin{aligned}\mathcal{R}_T(\mathcal{A}) = \frac{1}{T}\sum_{i\neq i^*}N_{i,T}\left[(\hat{\sigma}^2_{i,N_{i,T}}-\tilde{\sigma}^2_{i,N_{i,T}}) - \rho(\hat{\mu}_{i,N_{i,T}}-\tilde{\mu}_{i,N_{i,T}})\right] \\ + \frac{1}{T}\sum_{i=1}^{K}N_{i,T}\big(\hat{\mu}_{i,N_{i,T}}-\hat{\mu}_T(\mathcal{A})\big)^2 - \frac{1}{T}\sum_{i=1}^{K}N_{i,T}\big(\tilde{\mu}_{i,N_{i,T}}-\hat{\mu}_{i^*,T}\big)^2. \quad (3.12)\end{aligned}$$

If we further elaborate the second term, we obtain,

$$\begin{aligned}\frac{1}{T}\sum_{i=1}^{K}N_{i,T}\big(\hat{\mu}_{i,N_{i,T}}-\hat{\mu}_T(\mathcal{A})\big)^2 &= \frac{1}{T}\sum_{i=1}^{K}N_{i,T}\left(\hat{\mu}_{i,N_{i,T}}-\frac{1}{T}\sum_{j=1}^{K}N_{j,T}\hat{\mu}_{j,N_{j,T}}\right)^2 \\ &= \frac{1}{T}\sum_{i=1}^{K}N_{i,T}\left(\sum_{j=1}^{K}\frac{N_{j,T}}{T}(\hat{\mu}_{i,N_{i,T}}-\hat{\mu}_{j,N_{j,T}})\right)^2 \\ &\leq \frac{1}{T}\sum_{i=1}^{K}N_{i,T}\sum_{j=1}^{K}\frac{N_{j,T}}{T}(\hat{\mu}_{i,N_{i,T}}-\hat{\mu}_{j,N_{j,T}})^2 \\ &= \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i}N_{i,T}N_{j,T}(\hat{\mu}_{i,N_{i,T}}-\hat{\mu}_{j,N_{j,T}})^2.\end{aligned}$$

Using the definitions,

$$\widehat{\Delta}_i = (\hat{\sigma}^2_{i,N_{i,T}}-\tilde{\sigma}^2_{i,N_{i,T}}) - \rho(\hat{\mu}_{i,N_{i,T}}-\tilde{\mu}_{i,N_{i,T}}),$$

and

$$\widehat{\Gamma}^2_{i,j} = (\hat{\mu}_{i,N_{i,T}}-\hat{\mu}_{j,N_{j,T}})^2,$$

we finally obtain the following upper–bound on the regret,

$$\mathcal{R}_T(\mathcal{A}) \leq \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\widehat{\Delta}_i + \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\widehat{\Gamma}^2_{i,j}. \tag{3.13}$$

In the following, we rely on the terms,

$$\mathcal{R}_T^{\widehat{\Delta}} = \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\widehat{\Delta}_i$$

and

$$\mathcal{R}_T^{\widehat{\Gamma}} = \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\widehat{\Gamma}^2_{i,j}.$$

Unlike the definition in eq. 3.10, this upper bound explicitly illustrates the relationship between the regret and the number of pulls $N_{i,T}$; suggesting that a bound on the pulls is sufficient to bound the regret. This formulation also allows us to have a better understanding of how the regret is composed. Let consider the case of $\rho = 0$ (variance minimization problem). In this case, $\widehat{\Delta}_i$ represents the different in the empirical variances and $\widehat{\Gamma}_{i,j}$ is the difference in the empirical means. Even in a problem where all the arms have a zero variance (i.e., $\widehat{\Delta}_i = 0$), an algorithm pulling all the arms uniformly would suffer a constant regret due to the variance introduced by pulling arms with different means. Finally, we can also introduce a definition of the pseudo-regret.

Definition 6. *The pseudo regret for a learning algorithm $\mathcal{A}$ over T rounds is defined as,*

$$\widetilde{\mathcal{R}}_T(\mathcal{A}) = \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\Delta_i + \frac{2}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\Gamma^2_{i,j}, \tag{3.14}$$

where $\Delta_i = \mathrm{MV}_i - \mathrm{MV}_{i^*}$ and $\Gamma_{i,j} = \mu_i - \mu_j$.

In the following, we denote the two components of the pseudo–regret as,

$$\widetilde{\mathcal{R}}_T^{\Delta}(\mathcal{A}) = \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\Delta_i, \quad \text{and} \quad \widetilde{\mathcal{R}}_T^{\Gamma}(\mathcal{A}) = \frac{2}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\Gamma^2_{i,j}. \tag{3.15}$$

Where $\widetilde{\mathcal{R}}_T^{\Delta}(\mathcal{A})$ constitutes the standard regret derived from the traditional formulation of the multi-arm bandit problem and $\widetilde{\mathcal{R}}_T^{\Gamma}(\mathcal{A})$ denotes the exploration risk[5]. This regret can be shown to be close to the true regret up to small terms with high probability.

Lemma 1. *Given definitions 5 and 6,*

$$\mathcal{R}_T(\mathcal{A}) \leq \widetilde{\mathcal{R}}_T(\mathcal{A}) + (5+\rho)\sqrt{\frac{2K\log\left(\frac{1}{\delta}\right)}{T}} + 4\sqrt{2}\frac{K\log\left(\frac{1}{\delta}\right)}{T},$$

with probability at least $1 - 6nK\delta$.

The proof of Lemma 1 follows,

Proof. (Lemma 1)

We define a high–probability event in which the empirical values and the true values only differ for small quantities,

$$\mathcal{E} = \left\{\forall i = 1,\dots,K,\ \forall s = 1,\dots,T,\ \ \left|\hat{\mu}_{i,s} - \mu_i\right| \leq \sqrt{\frac{\log\frac{1}{\delta}}{2s}} \ \text{ and } \ \left|\hat{\sigma}^2_{i,s} - \sigma^2_i\right| \leq 5\sqrt{\frac{\log\frac{1}{\delta}}{2s}}\right\}.$$

Using Chernoff–Hoeffding inequality and a union bound over arms and rounds, we have that $\mathbb{P}[\mathcal{E}^C] \leq 6nK\delta$. Under this event, the empirical $\widehat{\Delta}_i$ can be upper–bounded by,

$$\begin{aligned}\widehat{\Delta}_i &= \Delta_i - (\sigma_i^2 - \sigma_{i^*}^2) + \rho(\mu_i - \mu_{i^*}) + (\hat{\sigma}^2_{i,N_{i,T}} - \tilde{\sigma}^2_{i,N_{i,T}}) - \rho(\hat{\mu}_{i,N_{i,T}} - \tilde{\mu}_{i,N_{i,T}}) \\ &\leq \Delta_i + 2(5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2N_{i,T}}},\end{aligned}$$

and similarly, $\widehat{\Gamma}_{i,j}$ can be upper–bounded by,

$$\begin{aligned}|\widehat{\Gamma}_{i,j}| &= |\Gamma_{i,j} - \mu_i + \mu_j + \hat{\mu}_{i,N_{i,T}} - \hat{\mu}_{j,N_{j,T}}| \\ &\leq |\Gamma_{i,j}| + \sqrt{\frac{\log\frac{1}{\delta}}{2N_{i,T}}} + \sqrt{\frac{\log\frac{1}{\delta}}{2N_{j,T}}}.\end{aligned}$$

[5]Notice that the factor 2 in front of the second term is due to a rough upper bounding used in the proof of Lemma 1.

Thus the regret can be written as,

$$\begin{aligned}
\mathcal{R}_T(\mathcal{A}) &\leq \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\left(\Delta_i + 2(5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2N_{i,T}}}\right) \\
&+ \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\left(|\Gamma_{i,j}| + \sqrt{\frac{\log\frac{1}{\delta}}{2N_{i,T}}} + \sqrt{\frac{\log\frac{1}{\delta}}{2N_{j,T}}}\right)^2 \\
&\leq \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\Delta_i + \frac{5+\rho}{T}\sum_{i\neq i^*}\sqrt{2N_{i,T}\log\frac{1}{\delta}} + \frac{2}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\Gamma_{i,j}^2 \\
&+ \frac{2\sqrt{2}}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{j,T}\log\frac{1}{\delta} + \frac{2\sqrt{2}}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}\log\frac{1}{\delta} \\
&\leq \frac{1}{T}\sum_{i\neq i^*} N_{i,T}\Delta_i + \frac{2}{T^2}\sum_{i=1}^{K}\sum_{j\neq i} N_{i,T}N_{j,T}\Gamma_{i,j}^2 + (5+\rho)\sqrt{\frac{2K\log\frac{1}{\delta}}{T}} + 4\sqrt{2}\frac{K\log\frac{1}{\delta}}{T}.
\end{aligned}$$

where in the next to last passage we used Jensen's inequality for concave functions and rough upper bounds on other terms ($K-1<K$, $\sum_{i\neq i^*} N_{i,T}\leq T$). By recalling the definition of $\widetilde{\mathcal{R}}_T(\mathcal{A})$, we finally obtain,

$$\mathcal{R}_T(\mathcal{A}) \leq \widetilde{\mathcal{R}}_T(\mathcal{A}) + (5+\rho)\sqrt{\frac{2K\log\frac{1}{\delta}}{T}} + 4\sqrt{2}\frac{K\log\frac{1}{\delta}}{T},$$

with probability $1-6nK\delta$. Thus we can conclude that any upper bound on the pseudo–regret $\widetilde{\mathcal{R}}_T(\mathcal{A})$ is a valid upper bound for the true regret $\mathcal{R}_T(\mathcal{A})$, up to a decreasing term of order $\mathcal{O}\left(\sqrt{\frac{K}{T}}\right)$.

□

The previous lemma shows that any (high–probability) bound on the pseudo–regret immediately translates into a bound on the true regret. Thus, we report most of the theoretical analysis according to $\widetilde{\mathcal{R}}_T(\mathcal{A})$. Nonetheless, it is interesting to notice the major difference between the true and pseudo–regret when compared to the standard bandit problem. In fact, it is possible to show in the risk-averse case that the pseudo–regret is not an unbiased estimator of the true regret, i.e., $\mathbb{E}[\mathcal{R}_T] \neq \mathbb{E}[\widetilde{\mathcal{R}}_T]$. Thus, in order to bound the expectation of $\mathcal{R}_T$ we build on the high–probability result from Lemma 1.

Input:

- Confidence δ
- Rounds T
- Arms K

For all $t = 1, \dots, T$**, repeat**

1. **For all** $t = 1, \dots, K$**, repeat**

$$B_{i,N_{i,t-1}} = \widehat{\mathrm{MV}}_{i,N_{i,t-1}} - (5+\rho)\sqrt{\frac{\log \frac{1}{\delta}}{2N_{i,t-1}}}$$

   end for
2. Learner chooses $I_t = \arg\min_{i=1,\dots,K} B_{i,N_{i,t-1}}$
3. Learner updates $N_{i,t} = N_{i,t-1} + 1$
4. Learner observes $X_{I_t,N_{i,t}} \sim \nu_{I_t}$
5. Learner updates $\widehat{\mathrm{MV}}_{i,N_{i,t}}$

end for

Figure 3.3: Pseudo-code of the *MV-LCB* algorithm.

4 Mean–Variance Lower Confidence Bound Algorithm

In this section we introduce a novel risk-averse bandit algorithm whose objective is to identify the arm which best trades off risk and return. The algorithm is a natural extension of *UCB1* [Auer, 2000] and we report a theoretical performance analysis on how well it balances the exploration needed to identify the best arm versus the risk of pulling arms with different means.

We propose an index–based bandit algorithm which estimates the Mean–Variance of each arm and selects the optimal arm according to the optimistic confidence–bounds on the current estimates. A sketch of the algorithm is reported in Figure 3.3. For each arm, the algorithm keeps track of the empirical Mean–Variance $\widehat{\mathrm{MV}}_{i,s}$ computed according to s samples. We can build high–probability confidence bounds on empirical Mean–Variance through an application of the Chernoff–Hoeffding inequality (see e.g., Antos et al. [2010] for the bound on the variance) on terms $\hat{\mu}$ and $\hat{\sigma}^2$.

Lemma 2. *Let* $\{X_{i,s}\}$ *be i.i.d. random variables bounded in* $[0,1]$ *from the distribution* ν_i *with mean* μ_i *and variance* σ_i^2*, and the empirical mean* $\hat{\mu}_{i,s}$ *and variance* $\hat{\sigma}_{i,s}^2$ *computed as in Equation 3.1, then,*

$$\mathbb{P}\left[\exists i = 1,\dots,K, s = 1,\dots,T, |\widehat{\mathrm{MV}}_{i,s} - \mathrm{MV}_i| \geq (5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2s}}\right] \leq 6nK\delta,$$

The algorithm in Figure 3.3 implements the principle of optimism in the face of uncertainty, where the algorithm computes upper confidence bounds for all the arms and chooses the arm with the highest bound. On the basis of the previous confidence bounds, we define a lower–confidence bound on the Mean–Variance of arm i when it has been pulled s times as,

$$B_{i,s} = \widehat{\mathrm{MV}}_{i,s} - (5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2s}}, \tag{3.16}$$

where δ is an input parameter of the algorithm. Given the index of each arm at each round t, the algorithm simply selects the arm with the smallest Mean–Variance index, i.e., $I_t = \arg\min_i B_{i,N_{i,t-1}}$. We refer to this algorithm as the Mean–Variance lower–confidence bound (*MV-LCB*) algorithm. We notice that the algorithm reduces to *UCB1* whenever $\rho \to \infty$. This is coherent with the fact that for $\rho \to \infty$ the Mean–Variance problem reduces to the maximization of the cumulative reward, for which *UCB1* is already known to be nearly-optimal. On the other hand, for $\rho = 0$, which leads to the problem of cumulative reward variance minimization, the algorithm plays according to a lower–confidence–bound on the variances.

The algorithm can also be easily improved by using tighter bounds on the mean and variance estimates. In particular, we can use Bernstein's inequality on the mean (see e.g., Audibert et al. [2009]) and a tighter deviation on the

variance Maurer and Pontil [2009], obtaining the index[6],

$$B^V_{i,s,t} = \left(\hat{\sigma}_{i,s} + \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2s}}\right)^2 - \rho\left(\hat{\mu}_{i,s} + \hat{\sigma}_{i,s}\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{s}} + \frac{\log\left(\frac{1}{\delta}\right)}{s}\right).$$

While this version of *MV-LCB* should work better whenever the variance of the arms is small, its theoretical properties would not differ much w.r.t. *MV-LCB* (see Audibert et al. [2009] for a comparison between *UCB-V* and *UCB*).

The *MV-LCB* algorithm is parameterized by a parameter δ which defines the confidence level of the bounds employed in the definition of the index (3.16). In Theorem 4 we show how to optimize the parameter when the horizon T is known in advance. On the other hand, if T is not known, it is possible to design an anytime version of *MV-LCB* by defining a non-decreasing exploration sequence $(\varepsilon_t)_t$ instead of the term $\log\frac{1}{\delta}$.

4.1 Theoretical Analysis

In this section we report the analysis of the regret $\mathcal{R}_T(\mathcal{A})$ of *MV-LCB* (Fig. 3.3). It is enough to analyze the number of pulls for each of the arms to recover a bound on the regret. The proofs are mostly based on similar arguments to the proof of *UCB*. We first report a high–probability bound on the number of pulls. The high–probability event over which the statement holds coincides with the event form of Lemma 1 which thus allows us to combine the two results and obtain a high–probability bound for the true regret $\mathcal{R}_T(\mathcal{A})$.

Lemma 3. *Let $b = 2(5+\rho)$, for any $\delta \in (0,1)$, the number of times each suboptimal arm $i \neq i^*$ is pulled by MV-LCB is,*

$$N_{i,T} \leq \frac{b^2}{\Delta_i^2}\log\frac{1}{\delta} + 1, \tag{3.17}$$

with probability of at least $1 - 6nK\delta$.

[6] We notice that in this case the estimated variance is computed as $\hat{\sigma}^2_{i,s} = \frac{1}{s-1}\sum_{s'=1}^{s} X^2_{i,s'} - \hat{\mu}^2_{i,s}$.

From the previous result, we derive the following regret bound in high probability and expectation.

Theorem 4. *Let the optimal arm i^* be unique and $b = 2(5+\rho)$, the MV-LCB algorithm achieves a pseudo–regret bounded as,*

$$\widetilde{\mathcal{R}}_T(\mathcal{A}) \leq \frac{b^2 \log \frac{1}{\delta}}{T}\left(\sum_{i\neq i^*}\frac{1}{\Delta_i} + 4\sum_{i\neq i^*}\frac{\Gamma^2_{i^*,i}}{\Delta_i^2} + \frac{2b^2\log\frac{1}{\delta}}{T}\sum_{i\neq i^*}\sum_{\substack{j\neq i\\ j\neq i^*}}\frac{\Gamma^2_{i,j}}{\Delta_i^2\Delta_j^2}\right) + \frac{5K}{T},$$

with probability at least $1-6nK\delta$. Similarly, if MV-LCB is run with $\delta = T^{-2}$ then,

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq \frac{2b^2 \log T}{T}\left(\sum_{i\neq i^*}\frac{1}{\Delta_i} + 4\sum_{i\neq i^*}\frac{\Gamma^2_{i^*,i}}{\Delta_i^2} + \frac{4b^2\log T}{T}\sum_{i\neq i^*}\sum_{\substack{j\neq i\\ j\neq i^*}}\frac{\Gamma^2_{i,j}}{\Delta_i^2\Delta_j^2}\right) + (17+6\rho)\frac{K}{T}.$$

Proof. (Lemma 3 and Theorem 4)

We begin by defining a high–probability event $\mathcal{E}$ as,

$$\mathcal{E} = \left\{\forall i = 1,\dots,K,\ \forall s = 1,\dots,T,\ \ \left|\hat{\mu}_{i,s} - \mu_i\right| \leq \sqrt{\frac{\log\frac{1}{\delta}}{2s}} \ \text{ and } \ \left|\hat{\sigma}^2_{i,s} - \sigma_i^2\right| \leq 5\sqrt{\frac{\log\frac{1}{\delta}}{2s}}\right\}.$$

Using Chernoff–Hoeffding inequality and a union bound over arms and rounds, we have that $\mathbb{P}[\mathcal{E}^C] \leq 6nK\delta$.

We now introduce the definition of the algorithm. Consider any time t when arm $i \neq i^*$ is pulled (i.e., $I_t = i$). By definition of the algorithm in Figure 3.3, i is selected if its corresponding index $B_{i,N_{i,t-1}}$ is bigger than for any other arm, notably the best arm i^*. By recalling the definition of the index and the empirical

Mean–Variance at time t, we have

$$\begin{aligned}\hat{\sigma}^2_{i,N_{i,t-1}} - \rho\hat{\mu}_{i,N_{i,t-1}} - (5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2N_{i,t-1}}} \\ &= B_{i,N_{i,t-1}} \\ &\leq B_{i^*,N_{i*,t-1}} \\ &= \hat{\sigma}^2_{i^*,N_{i^*,t-1}} - \rho\hat{\mu}_{i^*,N_{i^*,t-1}} - (5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2N_{i^*,t-1}}}.\end{aligned}$$

Over all the possible realizations, we now focus on the realizations in $\mathcal{E}$. In this case, we can rewrite the previous condition as,

$$\sigma_i^2 - \rho\mu_i - 2(5+\rho)\sqrt{\frac{\log\frac{1}{\delta}}{2N_{i,t-1}}} \leq B_{i,N_{i,t-1}} \leq B_{i^*,N_{i*,t-1}} \leq \sigma^2_{i^*} - \rho\mu_{i^*}.$$

Let time t be the last time when arm i is pulled until the final round T, then $N_{i,t-1} = N_{i,T} - 1$ and,

$$N_{i,T} \leq \frac{2(5+\rho)^2}{\Delta_i^2}\log\frac{1}{\delta} + 1,$$

which suggests that the suboptimal arms are pulled only few times with high probability. Plugging the bound in the regret in eq. 3.14 leads to the final statement,

$$\widetilde{\mathcal{R}}_T(\mathcal{A}) \leq \frac{1}{T}\sum_{i\neq i^*}\frac{b^2\log\frac{1}{\delta}}{\Delta_i} + \frac{1}{T}\sum_{i\neq i^*}\frac{4b^2\log\frac{1}{\delta}}{\Delta_i^2}\Gamma^2_{i^*,i} + \frac{1}{T^2}\sum_{i\neq i^*}\sum_{\substack{j\neq i\\ j\neq i^*}}\frac{2b^4(\log\frac{1}{\delta})^2}{\Delta_i^2\Delta_j^2}\Gamma^2_{i,j} + \frac{5K}{T},$$

with probability $1 - 6nK\delta$.

We now move from the previous high–probability bound to a bound in expectation. The pseudo–regret is (roughly) bounded as $\widetilde{\mathcal{R}}_T(\mathcal{A}) \leq 2 + \rho$ (by bounding $\Delta_i \leq 1 + \rho$ and $\Gamma \leq 1$), thus,

$$\begin{aligned}\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] &= \mathbb{E}\big[\widetilde{\mathcal{R}}_T(\mathcal{A})\mathbb{I}\{\mathcal{E}\}\big] + \mathbb{E}\big[\widetilde{\mathcal{R}}_T(\mathcal{A})\mathbb{I}\{\mathcal{E}^C\}\big] \\ &\leq \mathbb{E}\big[\widetilde{\mathcal{R}}_T(\mathcal{A})\mathbb{I}\{\mathcal{E}\}\big] + (2+\rho)\mathbb{P}\big[\mathcal{E}^C\big].\end{aligned}$$

By using the previous high–probability bound and recalling that $\mathbb{P}[\mathcal{E}^C] \leq 6nK\delta$, we have,

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq \frac{1}{T}\sum_{i\neq i^*}\frac{b^2\log\frac{1}{\delta}}{\Delta_i} + \frac{1}{T}\sum_{i\neq i^*}\frac{4b^2\log\frac{1}{\delta}}{\Delta_i^2}\Gamma_{i^*,i}^2 + \frac{1}{T^2}\sum_{i\neq i^*}\sum_{\substack{j\neq i\\ j\neq i^*}}\frac{2b^4(\log\frac{1}{\delta})^2}{\Delta_i^2\Delta_j^2}\Gamma_{i,j}^2 + \frac{5K}{T} + (2+\rho)6nK\delta.$$

The final statement of the lemma follows by tuning the parameter $\delta = T^{-2}$ so as to have a regret bound decreasing with T. □

While a high–probability bound for $\mathcal{R}_T$ can be immediately obtained from Lemma 1, the expectation of $\mathcal{R}_T$ is reported in the next proof.

Proof. Since the Mean–Variance $-\rho \leq \widehat{\mathrm{MV}} \leq \frac{1}{4}$, the regret is bounded by $-\frac{1}{4}-\rho \leq \mathcal{R}_T(\mathcal{A}) \leq \frac{1}{4}+\rho$. Thus we have,

$$\mathbb{E}[\mathcal{R}_T(\mathcal{A})] \leq u\mathbb{P}[\mathcal{R}_T(\mathcal{A}) \leq u] + \left(\frac{1}{4}+\rho\right)\mathbb{P}[\mathcal{R}_T(\mathcal{A}) > u].$$

By taking u equal to the previous high–probability bound and recalling that $\mathbb{P}[\mathcal{E}^C] \leq 6nK\delta$, we have,

$$\mathbb{E}[\mathcal{R}_T(\mathcal{A})] \leq \frac{1}{T}\sum_{i\neq i^*}\frac{b^2\log\frac{1}{\delta}}{\Delta_i} + \frac{1}{T}\sum_{i\neq i^*}\frac{4b^2\log\frac{1}{\delta}}{\Delta_i^2}\Gamma_{i^*,i}^2 + \frac{1}{T^2}\sum_{i\neq i^*}\sum_{\substack{j\neq i\\ j\neq i^*}}\frac{2b^4(\log\frac{1}{\delta})^2}{\Delta_i^2\Delta_j^2}\Gamma_{i,j}^2 + \frac{5K}{T} + b\sqrt{\frac{K\log\frac{1}{\delta}}{2n}} + 4\sqrt{2}\frac{K\log\frac{1}{\delta}}{T} + \left(\frac{1}{4}+\rho\right)6nK\delta.$$

The final statement of the lemma follows by tuning the parameter $\delta = T^{-2}$ so as to have a regret bound decreasing with T. □

Bound

Let $\Delta_{\min} = \min_{i\neq i^*}\Delta_i$ and $\Gamma_{\max} = \max_i|\Gamma_i|$, then a rough simplification of the

previous bound leads to,

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq \mathcal{O}\left(\frac{K}{\Delta_{\min}}\frac{\log T}{T} + K^2\frac{\Gamma^2_{\max}}{\Delta^4_{\min}}\frac{\log^2 T}{T^2}\right).$$

First we notice that the regret decreases as $\mathcal{O}\left(\frac{\log T}{T}\right)$, implying that *MV-LCB* is a consistent algorithm. As already highlighted in Definition 5, the regret is mainly composed by two terms. The first term is due to the difference in the Mean–Variance of the best arm and the arms pulled by the algorithm, while the second term denotes the additional variance introduced by the exploration risk of pulling arms with different means. In particular, it is interesting to note that this additional term depends on the squared difference in the means of the arms $\Gamma^2_{i,j}$. Thus, if all the arms have the same mean, this term would be zero.

4.2 Worst–Case Analysis

We can further study the result of Theorem 4 by considering the worst–case performance of *MV-LCB*, that is the performance when the distributions of the arms are chosen so as to maximize the regret. In order to illustrate our argument we consider the simple case of $K = 2$ arms, $\rho = 0$ (variance minimization), $\mu_1 \neq \mu_2$, and $\sigma_1^2 = \sigma_2^2 = 0$ (deterministic arms)[7]. In this case we have a variance gap $\Delta = 0$ and $\Gamma^2 > 0$. According to the definition of *MV-LCB*, the index $B_{i,s}$ would simply reduce to,

$$B_{i,s} = \sqrt{\frac{\log\frac{1}{\delta}}{s}},$$

thus forcing the algorithm to pull both arms uniformly (i.e., $N_{1,T} = N_{2,T} = \frac{T}{2}$ up to rounding effects). Since the arms have the same variance, there is no direct regret in pulling either one or the other. Nonetheless, the algorithm has an additional variance due to the difference in the samples drawn from distributions with

[7]Note that in this case (i.e., $\Delta = 0$), Theorem 4 does not hold, since the optimal arm is not unique.

different means. In this case, the algorithm suffers a constant (true) regret,

$$\begin{aligned}\mathcal{R}_T(\textit{MV-LCB}) &= \frac{N_{1,T}N_{2,T}}{T^2}\Gamma^2 \\ &= \frac{1}{4}\Gamma^2,\end{aligned}$$

independent from the number of rounds T. This argument can be generalized to multiple arms and $\rho \neq 0$, since it is always possible to design an environment (i.e., a set of distributions) such that $\Delta_{\min} = 0$ and $\Gamma_{\max} \neq 0$[8]. This result is not surprising. In fact, two arms with the same Mean–Variance are likely to produce similar observations, thus leading *MV-LCB* to pull the two arms repeatedly over time, since the algorithm is designed to try to discriminate between similar arms. Although this behavior does not suffer from any regret in pulling the "suboptimal" arm (the two arms are equivalent), it does introduce an additional variance, due to the difference in the means of the arms ($\Gamma \neq 0$), which finally leads to a regret the algorithm is not "aware" of. This argument suggests that, for any T, it is always possible to design an environment for which *MV-LCB* has a constant regret. This is particularly interesting since it reveals a huge gap between the Mean–Variance problem and the standard expected regret minimization problem and will be further investigated in the numerical simulations presented in Section 6. In fact, in the latter case, *UCB* is known to have a worst–case regret per round of $\Omega\left(T^{-\frac{1}{2}}\right)$ [Audibert et al., 2010], while in the worst case, *MV-LCB* suffers a constant regret. In the next section we introduce a simple algorithm able to deal with this problem and achieve a vanishing worst–case regret.

5 Exploration–Exploitation Algorithm

Although for any fixed problem (with $\Delta_{\min} > 0$) the *MV-LCB* algorithm has a vanishing regret, for any value of T, it is always possible to find an environment for which its regret is constant. In this section, we analyze a simple algorithm where exploration and exploitation are two distinct phases. As shown in Figure 3.4,

[8]Notice that this is always possible for a large majority of distributions for which the mean and variance are independent or mildly correlated.

Input: Length of the exploration phase τ, Rounds T, Arms K
Exploration Phase
For all $t = 1, \dots, \frac{\tau}{K}$**, repeat**

For all $t = 1, \dots, K$**, repeat**

Learner observes $X_{i,t} \sim \nu_i$

end for

end for
Learner computes the estimates $\widehat{\mathrm{MV}}_{i,\frac{\tau}{K}}$
Learner computes $\hat{i}^* = \arg\min_i \widehat{\mathrm{MV}}_{i,\frac{\tau}{K}}$
Exploitation Phase
For all $t = \tau + 1, \dots, T$**, repeat**

Learner selects $\hat{i}^*$

end for

Figure 3.4: Pseudo-code of the *ExpExp* algorithm.

the *ExpExp* algorithm divides the time horizon T into two distinct phases of length τ and $T - \tau$ respectively. During the first phase all the arms are explored uniformly, thus collecting $\frac{\tau}{K}$ samples each[9]. Once the exploration phase is over, the Mean–Variance of each arm is computed and the arm with the smallest estimated Mean–Variance $\widehat{\mathrm{MV}}_{i,\frac{\tau}{K}}$ is repeatedly pulled until the end.

5.1 Theoretical Analysis

The *MV-LCB* is specifically designed to minimize the probability of pulling the wrong arms, so whenever there are two equivalent arms (i.e., arms with the same Mean–Variance), the algorithm tends to pull them the same number of times, at the cost of potentially introducing an additional variance which might result in a constant regret. On the other hand, *ExpExp* stops exploring the arms after τ rounds and then elicits one arm as the best and keeps pulling it for the remaining $T - \tau$ rounds. Intuitively, the parameter τ should be tuned so as to meet different requirements. The first part of the regret (i.e., the regret coming from pulling the suboptimal arms) suggests that the exploration phase τ should be long enough for the algorithm to select the empirically best arm $\hat{i}^*$ at τ equivalent to the actual

[9]In the definition and in the following analysis we ignore rounding effects.

optimal arm i^* with high probability; and at the same time, as short as possible to reduce the number of times the suboptimal arms are explored. On the other hand, the second part of the regret (i.e., the variance of pulling arms with different means) is minimized by taking τ as small as possible (e.g., $\tau = 0$ would guarantee a zero regret). During the exploitation phase the algorithm pulls arm $\hat{i}^*$ with the smallest empirical variance estimated during the exploration phase of length τ. As a result, the number of pulls of each arm is,

$$N_{i,T} = \frac{\tau}{K} + (T - \tau)\mathbb{I}\{i = \hat{i}^*\} \tag{3.18}$$

We analyze the two terms of the regret separately, where,

$$\begin{aligned}\widetilde{\mathcal{R}}_T^{\Delta} &= \frac{1}{T}\sum_{i\neq i^*}\left(\frac{\tau}{K} + (T-\tau)\mathbb{I}\{i = \hat{i}^*\}\right)\Delta_i \\ &= \frac{\tau}{nK}\sum_{i\neq i^*}\Delta_i + \frac{T-\tau}{T}\sum_{i\neq i^*}\underbrace{\Delta_i\mathbb{I}\{i=\hat{i}^*\}}_{(a)}.\end{aligned}$$

The following theorem illustrates the optimal trade-off between these terms.

Theorem 5. *Let ExpExp be run with $\tau = K\left(\frac{T}{14}\right)^{2/3}$, then for any choice of distributions $\{\nu_i\}$ the expected regret is,*

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq 2\frac{K}{T^{1/3}}. \tag{3.19}$$

Proof. We notice that the only random variable in this formulation is the best arm $\hat{i}^*$ at the end of the exploration phase. We thus compute the expected value

of $\widetilde{\mathcal{R}}_T^{\Delta}$, where,

$$\begin{aligned}
\mathbb{E}[(a)] &= \mathbb{P}[i = \hat{i}^*]\Delta_i \\
&= \mathbb{P}[\forall j \neq i,\ \hat{\sigma}^2_{i,\tau/K} \leq \hat{\sigma}^2_{j,\tau/K}]\Delta_i \\
&\leq \mathbb{P}[\hat{\sigma}^2_{i,\tau/K} \leq \hat{\sigma}^2_{i^*,\tau/K}]\Delta_i \\
&= \mathbb{P}\big[(\hat{\sigma}^2_{i,\tau/K} - \sigma_i^2) + (\sigma_{i^*}^2 - \hat{\sigma}^2_{i^*,\tau/K}) \leq \Delta_i\big]\Delta_i \\
&\leq 2\Delta_i \exp\left(-\frac{\tau}{K}\Delta_i^2\right).
\end{aligned}$$

The second term in the regret can be bounded as follows.

$$\begin{aligned}
\widetilde{\mathcal{R}}_T^{\Gamma} &= \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i}\left(\frac{\tau}{K} + (T-\tau)\mathbb{I}\{i=\hat{i}^*\}\right)\left(\frac{\tau}{K} + (T-\tau)\mathbb{I}\{j=\hat{i}^*\}\right)\Gamma^2_{i,j} \\
&= \frac{1}{T^2}\sum_{i=1}^{K}\sum_{j\neq i}\left(\frac{\tau^2}{K^2} + (T-\tau)^2\mathbb{I}\{i=\hat{i}^*\}\mathbb{I}\{j=\hat{i}^*\} + \frac{\tau}{K}(T-\tau)\mathbb{I}\{j=\hat{i}^*\} + \frac{\tau}{K}(T-\tau)\mathbb{I}\{i=\hat{i}^*\}\right)\Gamma \\
&= \frac{\tau^2}{T^2K^2}\sum_{i=1}^{K}\sum_{j\neq i}\Gamma^2_{i,j} + 2\frac{(T-\tau)\tau}{KT^2}\sum_{i=1}^{K}\sum_{j\neq i}\Gamma^2_{i,j}\mathbb{I}\{i=\hat{i}^*\} \\
&\leq \frac{\tau}{T^2} + 2\frac{(T-\tau)\tau}{T^2} \leq 2\frac{\tau}{T}.
\end{aligned}$$

Grouping all the terms, *ExpExp* has an expected regret bounded as,

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq 2\frac{\tau}{T} + 2\sum_{i\neq i^*}\Delta_i \exp\left(-\frac{\tau}{K}\Delta_i^2\right).$$

We can now move to the worst–case analysis of the regret. Let $f(\Delta_i) = \Delta_i \exp\left(-\frac{\tau}{K}\Delta_i^2\right)$, the "adversarial" choice of the gap is determined by maximizing the regret, which corresponds to,

$$\begin{aligned}
f'(\Delta_i) &= \exp\left(-\frac{\tau}{K}\Delta_i^2\right) + \Delta_i\left(-2\frac{\tau}{K}\Delta_i\exp\left(-\frac{\tau}{K}\Delta_i^2\right)\right) \\
&= \left(1 - 2\frac{\tau}{K}\Delta_i^2\right)\exp\left(-\frac{\tau}{K}\Delta_i^2\right) \\
&= 0,
\end{aligned}$$

and leads to a worst–case choice for the gap of,

$$\Delta_i = \sqrt{\frac{K}{2\tau}}.$$

The worst–case regret is then,

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq 2\frac{\tau}{T} + (K-1)\sqrt{2K}\frac{1}{\sqrt{\tau}}\exp(-0.5) \leq 2\frac{\tau}{T} + K^{3/2}\frac{1}{\sqrt{\tau}}.$$

We can now choose the parameter τ minimizing the worst–case regret. Taking the derivative of the regret w.r.t. τ we obtain,

$$\frac{d\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})]}{d\tau} = \frac{2}{T} - \frac{1}{2}\left(\frac{K}{\tau}\right)^{3/2} = 0,$$

thus leading to the optimal parameter $\tau = \left(\frac{T}{4}\right)^{2/3} K$. The final regret is thus bounded as,

$$\mathbb{E}[\widetilde{\mathcal{R}}_T(\mathcal{A})] \leq 3\frac{K}{T^{1/3}}.$$

□

We first notice that this bound suggests that *ExpExp* performs worse than *MV-LCB* on easy problems. In fact, Theorem 4 demonstrates that *MV-LCB* has a regret decreasing as $\mathcal{O}\left(K\frac{\log(T)}{T}\right)$ whenever the gaps Δ are not small compared to T, while in the remarks of Theorem 4 we highlighted the fact that for any value of T, it is always possible to design an environment which leads *MV-LCB* to suffer a constant regret. On the other hand, the previous bound for *ExpExp* is distribution independent and indicates the regret is still a decreasing function of T even in the worst case. This opens the question whether it is possible to design an algorithm which works as well as *MV-LCB* on easy problems and as robustly as *ExpExp* on difficult problems.

The previous result can be improved by changing the exploration strategy used in the first τ rounds. Instead of a pure uniform exploration of all the arms,

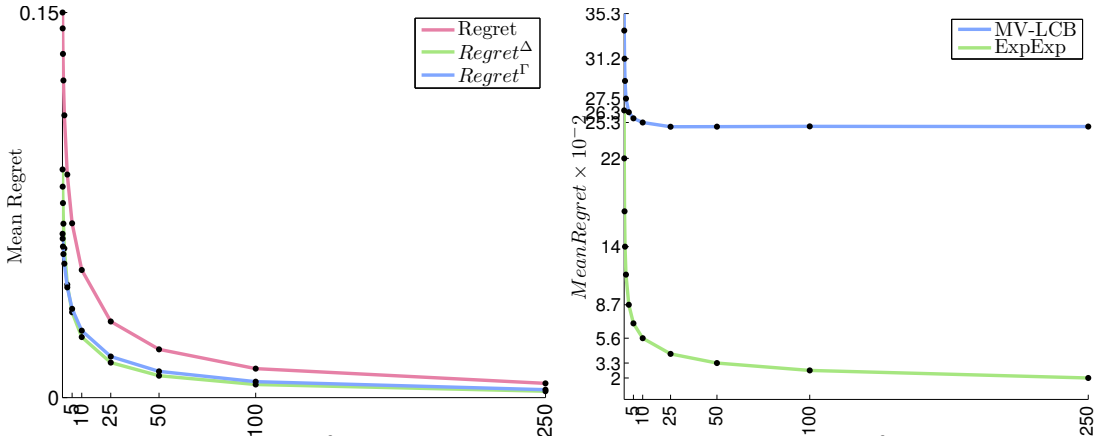

Figure 3.5: *MV-LCB* Regret (LEFT) and worst-case performance of *MV-LCB* versus *ExpExp*, for different values of $T \times 10^3$(RIGHT).

we could adopt a best–arm identification algorithms such as *Successive Reject* or *UCB-E*, which maximize the probability of returning the best arm given a fixed budget of rounds τ (see e.g., Audibert et al. [2010]).

6 Numerical Simulations

In this section we report numerical simulations aimed at validating the main theoretical findings reported in the previous sections. In the following graphs we study the true regret $\mathcal{R}_T(\mathcal{A})$ averaged over 500 runs. We first consider the variance minimization problem ($\rho = 0$) with $K = 2$ Gaussian arms set to $\mu_1 = 1.0$, $\mu_2 = 0.5$, $\sigma_1^2 = 0.05$, and $\sigma_2^2 = 0.25$ and run *MV-LCB*[10]. In Figure 3.5 we report the true regret $\mathcal{R}_T$ (as in the original definition in eq. 3.12) and its two components $\mathcal{R}_T^{\widehat{\Delta}}$ and $\mathcal{R}_T^{\widehat{\Gamma}}$ (these two values are defined as in eq. 3.15 with $\widehat{\Delta}$ and $\widehat{\Gamma}$ replacing Δ and Γ). As expected (see e.g., Theorem 4), the regret is characterized by the regret realized from pulling suboptimal arms and arms with different means (Exploration Risk) and tends to zero as T increases. Indeed, if we considered two distributions with equal means ($\mu_1 = \mu_2$), the average regret coincides with $\mathcal{R}_T^{\widehat{\Delta}}$. Furthermore, as shown in Theorem 4 the two regret terms decrease with the same rate $\mathcal{O}\left(\frac{\log T}{T}\right)$.

[10]Notice that although in this chapter we assumed the distributions to be bounded in $[0, 1]$ all the results can be extended to sub-Gaussian distributions.

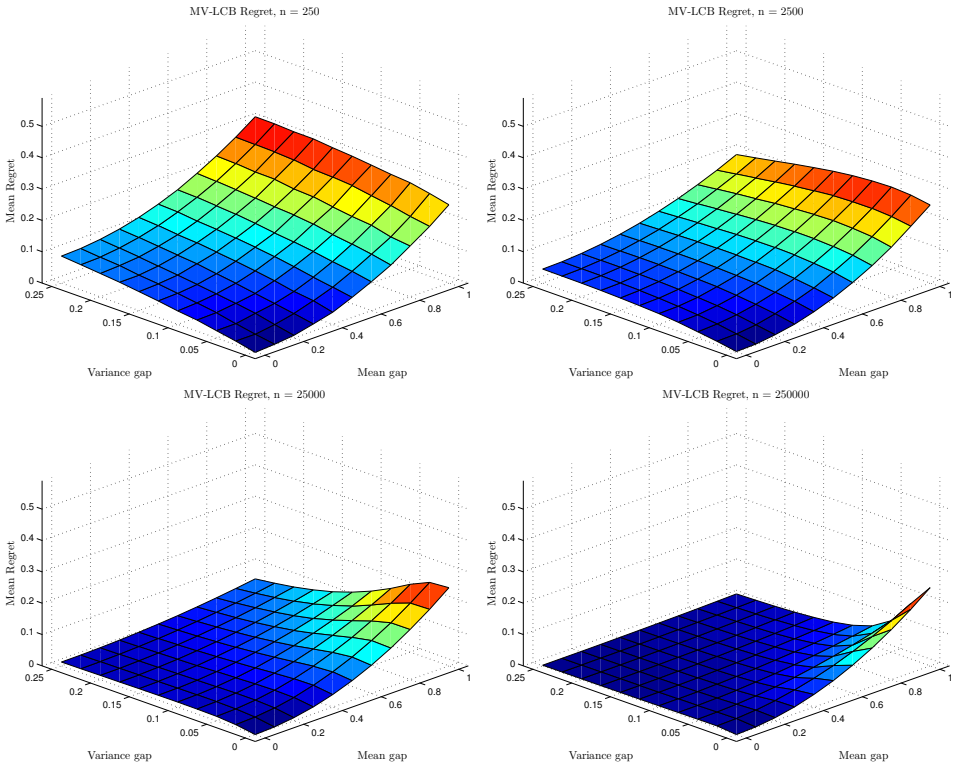

Figure 3.6: Regret $\mathcal{R}_T$ of *MV-LCB*.

7 Sensitivity Analysis

Here we detail the impact of Δ and Γ on the performance of *MV-LCB* and compare the worst–case performance of *MV-LCB* to *ExpExp* (see Figure 3.5). In order to have a fair comparison, for any value of T and for each of the two algorithms, we select the pair Δ_w, Γ_w which corresponds to the largest regret (we search in a grid of values with $\mu_1 = 1.5$, $\mu_2 \in [0.4; 1.5]$, $\sigma_1^2 \in [0.0; 0.25]$, and $\sigma_2^2 = 0.25$, so that $\Delta \in [0.0; 0.25]$ and $\Gamma \in [0.0; 1.1]$). As discussed in Section 5, while the worst–case regret of *ExpExp* keeps decreasing over T, it is always possible to find a problem for which regret of *MV-LCB* stabilizes to a constant.

We consider the variance minimization problem ($\rho = 0$) with $K = 2$ Gaussian arms with different means and variances. In particular, we consider a grid of values with $\mu_1 = 1.5$, $\mu_2 \in [0.4; 1.5]$, $\sigma_1^2 \in [0.0; 0.25]$, and $\sigma_2^2 = 0.25$, so that $\Delta \in [0.0; 0.25]$ and $\Gamma \in [0.0; 1.1]$ and number of rounds $T \in [50; 2.5 \times 10^5]$. Figures 3.6 and 3.7 report the mean regret for different values of T. The colors are renormalized in

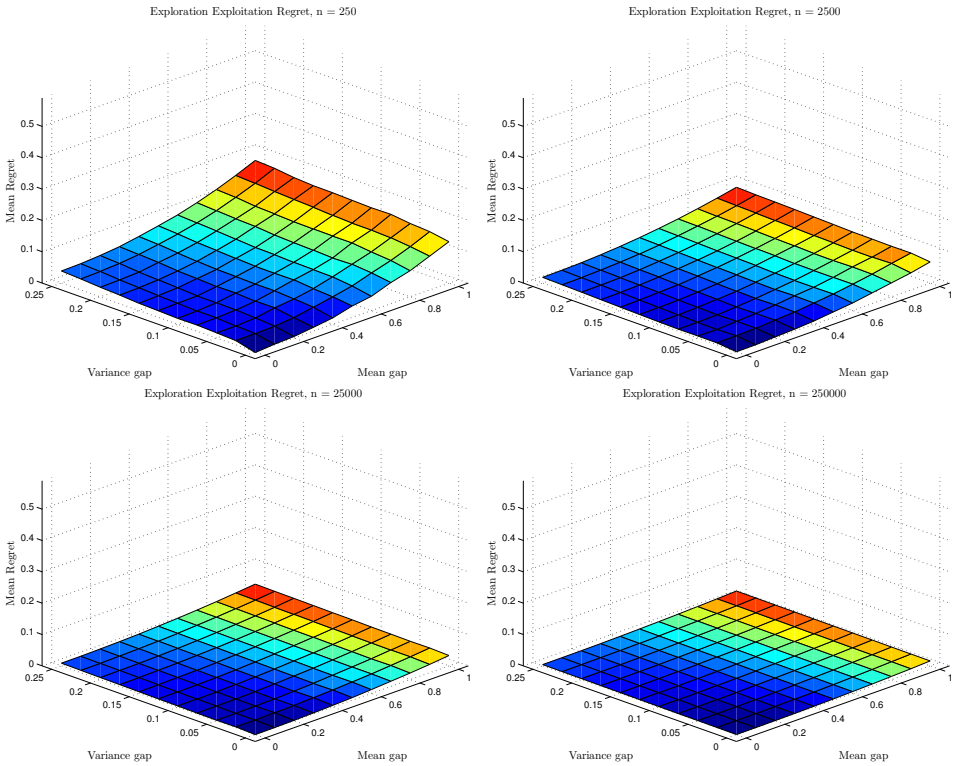

Figure 3.7: Regret $\mathcal{R}_T$ of *ExpExp*.

each plot so that dark blue corresponds to the smallest regret and red to the largest regret. The results confirm the theoretical findings of Theorem 4 and 5. In fact, for simple problems (large gaps Δ) *MV-LCB* converges to a zero–regret faster than *ExpExp*, while for Δ close to zero (i.e., equivalent arms), *MV-LCB* has a constant regret which does not decrease with T and the regret of *ExpExp* slowly decreases to zero.

In Section 6 we report numerical results demonstrating the composition of the regret and performance of algorithms with only two arms in the case of variance minimization. Here we report results for a wide range of risk tolerance $\rho \in [0.0; 10.0]$ and $K = 15$ arms. We set the mean and variance for each of the 15 arms so that a subset of arms is always dominated (i.e., for any ρ, $\mathrm{MV}_i^\rho > \mathrm{MV}_{i_\rho^*}^\rho$) demonstrating the effect of different ρ values on the position of the optimal arm i_ρ^*.

In Figure 3.5 we arranged the true values of each arm along the red frontier and the ρ-directed performance of the algorithms in a standard deviation–mean

Arm	μ	σ^2
α_1	0.10	0.05
α_2	0.20	0.34
α_3	0.23	0.28
α_4	0.27	0.09
α_5	0.32	0.23
α_6	0.32	0.72
α_7	0.34	0.19
α_8	0.41	0.14
α_9	0.43	0.44
α_{10}	0.54	0.53
α_{11}	0.55	0.24
α_{12}	0.56	0.36
α_{13}	0.67	0.56
α_{14}	0.71	0.49
α_{15}	0.79	0.85

Arm	μ	σ^2
α_1	0.1	0.05
α_2	0.2	0.0725
α_3	0.27	0.09
α_4	0.32	0.11
α_5	0.41	0.145
α_6	0.49	0.19
α_7	0.55	0.24
α_8	0.59	0.28
α_9	0.645	0.36
α_{10}	0.678	0.413
α_{11}	0.69	0.445
α_{12}	0.71	0.498
α_{13}	0.72	0.53
α_{14}	0.765	0.72
α_{15}	0.79	0.854

Figure 3.8: Configuration 1 and Configuration 2.

plot. The green and blue lines show the standard deviation and mean for the performance of each algorithm for a specific ρ setting and fixed horizon T, where each point represents the resulting mean–standard deviation of the sequence of pulls on the arms by the algorithm with that specific value of ρ. The gap between the ρ specific performance of the algorithm and the corresponding optimal arm along the red frontier represents the regret for the specific ρ value. Accordingly, the gap between the algorithm performance curves represents the gap in performance with regard to *MV-LCB* versus *ExpExp*. Where a lot of arms have big gaps (e.g., all the dominated arms have a large gap for any value of ρ), *MV-LCB* tends to perform better than *ExpExp*. The series of plots represent increasing values of T and demonstrate the relative algorithm performance versus the optimal red frontier. The set of plots represent the two settings reported in Figure 3.8. We chose the values of the arms so as to have configurations with different complexities. In particular, configuration 1 corresponds to "easy" problems for *MV-LCB* since the arms all have quite large gaps (for different values of ρ) and this should allow it to perform well. On the other hand, the second configuration has much smaller gaps and, thus, higher complexity. According to the bounds for *MV-LCB* we know that a good proxy for its learning complexity is represented by the term $\sum_i \frac{1}{\Delta^2_{i,\rho}}$.

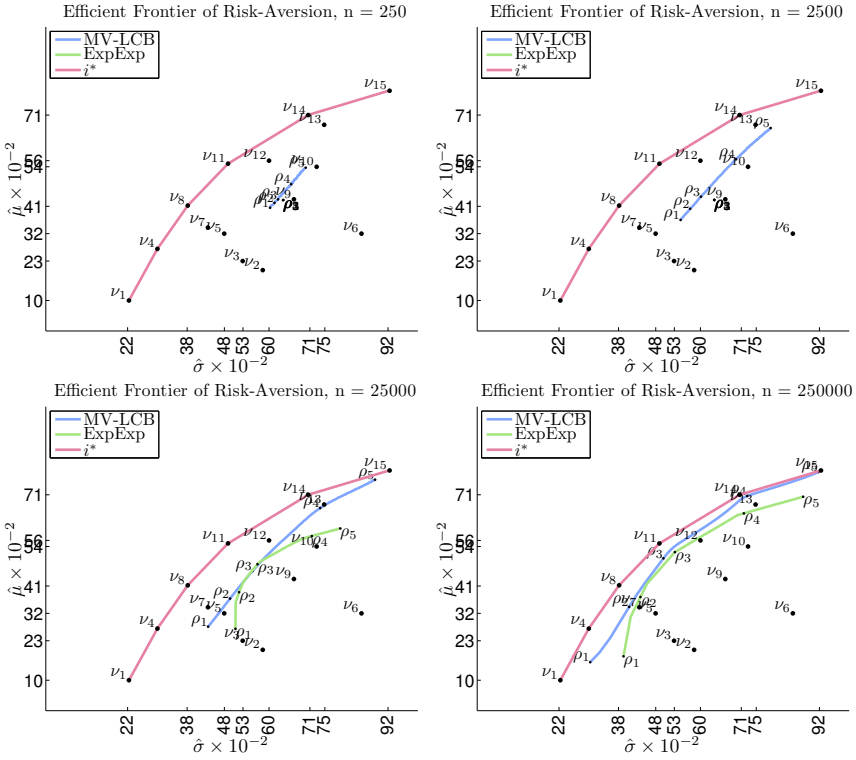

Figure 3.9: Risk tolerance sensitivity of *MV-LCB* and *ExpExp* for *Configuration 1.*

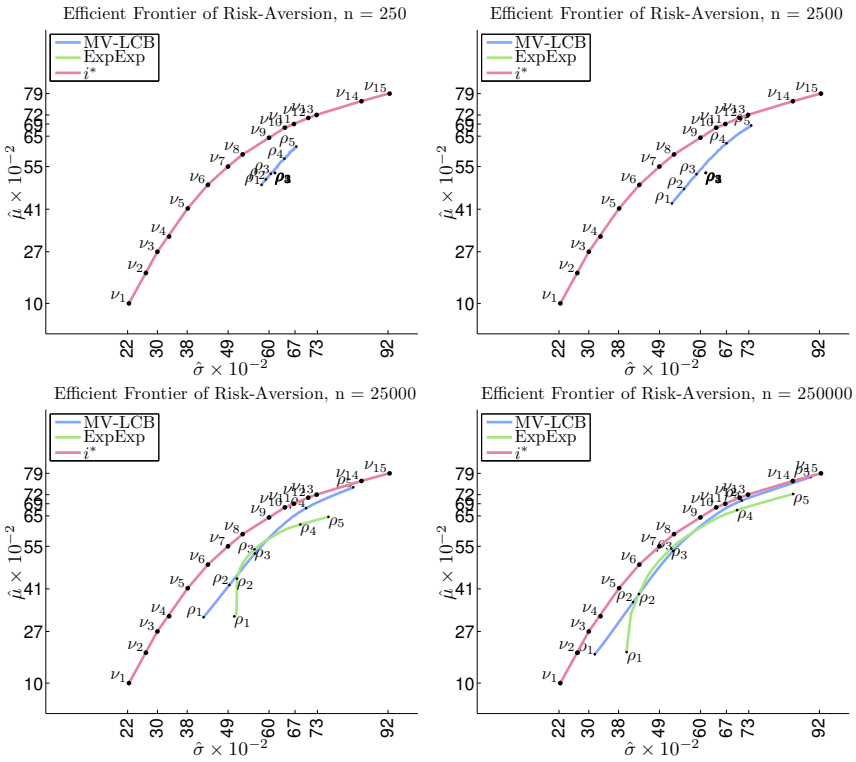

Figure 3.10: Risk tolerance sensitivity of *MV-LCB* and *ExpExp* for *Configuration 2.*

As we notice, in both configurations the performance of *MV-LCB* and *ExpExp* approach one of the optimal arms i^*_ρ for each specific ρ as T increases. Nonetheless, in configuration 1 the large number of suboptimal arms (e.g., arms with large gaps) allows *MV-LCB* to outperform *ExpExp* and converge faster to the optimal arm (and thus zero regret). On the other hand, in configuration 2 there are more arms with similar performance and for some values of ρ *ExpExp* eventually achieves better performance than *MV-LCB*.

8 Discussion

In this chapter, we evaluate the *risk* of an algorithm in terms of the variability of the sequences of samples that it actually generates. Although this notion might resemble other analyses of UCB-based algorithms (see e.g., the high-probability analysis in Audibert et al. [2009]), it captures different features of the learning algorithm. Whenever a bandit algorithm is run over T rounds, its behavior, combined with the arms' distributions, generates a probability distribution over sequences of T rewards. While the *quality* of this sequence is usually defined by its cumulative sum (or average), here we say that a sequence of rewards is *good* if it displays a good trade-off between its (empirical) mean and variance. It is important to notice that this notion of risk-return trade–off does not coincide with the variance of the algorithm over multiple runs.

Let us consider a simple case with two arms that deterministically generate 0s and 1s respectively, and two different algorithms. Algorithm $\mathcal{A}_1$ pulls the arms in a fixed sequence at each run (e.g., arm 1, arm 2, arm 1, arm 2, and so on), so that each arm is always pulled $\frac{T}{2}$ times. Algorithm $\mathcal{A}_2$ chooses one arm uniformly at random at the beginning of the run and repeatedly pulls this arm for T rounds. Algorithm $\mathcal{A}_1$ generates sequences such as 010101... which have high variability within each run, incurs a high regret (e.g., if $\rho = 0$), but has no variance over multiple runs because it always generates the same sequence. On the other hand, $\mathcal{A}_2$ has no variability in each run, since it generates sequences with only 0s or only 1s, suffers no regret in the case of variance minimization, but has high variance

over multiple runs since the two completely different sequences are generated with equal probability. This simple example demonstrates that an algorithm with a very small standard regret w.r.t. the cumulative reward (e.g., $\mathcal{A}_1$), might result in a very high variability in a single run of the algorithm, while an algorithm with small mean-variance regret (e.g., $\mathcal{A}_2$) could have a high variance over multiple runs.

9 Conclusions

The majority of multi–armed bandit literature focuses on the minimizing the regret w.r.t. the arm with the highest return in expectation. In this chapter, we introduced a novel multi–armed bandit setting where the objective is to perform as well as the arm with the best risk–return trade–off. In particular, we relied on the Mean–Variance objective introduced in Markowitz [1952] to measure the performance of the arms and to define the regret of a learning algorithm. The impact of this particular risk objective is the need to manage variance over multiple runs versus the variability over a single run. The later case highlights an interesting effect on the regret. Decision-making, while managing the variability within a single sequence, is tricky. In particular, controlling the variance over multiple runs does not necessarily control the risk of variability over a single run. We proposed two novel algorithms to solve the Mean–Variance bandit problem and we reported their corresponding theoretical analysis. While *MV-LCB* shows a small regret of order $\mathcal{O}\left(\frac{\log T}{T}\right)$ on "easy" problems (i.e., where the Mean–Variance gaps Δ are big w.r.t. T), we showed that it has a constant worst–case regret. On the other hand, we proved that *ExpExp* has a vanishing worst–case regret at the cost of worse performance on "easy" problems. To the best of our knowledge this is the first work introducing risk–aversion in the multi–armed bandit setting and it opens a series of interesting questions.

Lower–bound. *MV-LCB* has a regret of order $\mathcal{O}\left(\sqrt{\frac{K}{T}}\right)$ on easy problems and $\mathcal{O}(1)$ on difficult problems, while *ExpExp* achieves the same regret $\mathcal{O}\left(KT^{-\frac{1}{3}}\right)$ over all problems. The primary open question is whether $\mathcal{O}\left(KT^{-\frac{1}{3}}\right)$ is actually

the best possible achievable rate (in the worst–case) for this problem or a better rate is possible. This question is of particular interest since the standard reward expectation maximization problem has a known lower–bound of $\Omega\left(T^{-\frac{1}{2}}\right)$, while the minimax rate of $\Omega\left(T^{-\frac{1}{3}}\right)$ for the Mean–Variance problem would imply that the risk-averse bandit problem is intrinsically more difficult than the standard bandit problem.

Multi-period Risk. The notion of optimality in a risk sensitive setting depends on the best sequence of arms. Under a Mean–Variance objective, the best sequence of arms corresponds to the best single arm, so both the single and multi-period cases happen to coincide. This is not necessarily the case for other popular measures of risk, such as the conditional–value–at–risk or value–at–risk. In particular, the optimal single-period risk corresponds to a single arm, while the optimal multi-period risk sequence of choices is defined by the minimum risk over the best sequence of arms. In the case of the standard expectation maximization setting, the cumulative expected reward is simply the sum of single-period expected rewards by linearity of expectation. Under a risk objective, where objectives are mostly nonlinear, risk does not typically decompose into a sum over single-period risks. As a result, evaluating arms according to their single-period risk does not imply a correct preference with respect to a multi-period risk objective. For example, the variance of the sum of T independent realizations of the same random variable is simply T times its variance. For other measures of risk (e.g., α value–at–risk), this is not necessarily the case. As a result, an arm with the smallest single-period risk might not be the optimal choice over a horizon of T rounds. Therefore, the performance of a learning algorithm should be compared to the smallest risk that can be achieved by any sequence of arms over T rounds, thus requiring a new definition of regret.

Alternative risk statistics. There are several alternative notions of risk that are straightforward extensions to this work. In fact, while the cumulative distribution of a random variable can be reliably estimated (see e.g., Massart [1990]), estimating the quantile might be more difficult. In Artzner et al. [1999], axiomatic rules are listed to define coherent measures of risk. Though α value–at–risk violates these

rules, conditional value at risk passes these rules as a coherent measure of risk. One can easily imagine a lower confidence bound algorithm based on Brown [2007] in the same composition as *MV-LCB* which replaces the variance by the conditional value at risk.

10 Subsequent Work

Subsequent to the introduction of risk objectives to the stochastic multi-arm bandit problem in the original publication of this work, several additional works have been published.

Galichet et al. [2013] considers an objective defined by the conditional value at risk, a coherent measures of risk [Artzner et al., 1999]. Their focus is on applications where the exploration of the environment is risky, with the aim of learning a policy that trades-off between exploration, exploitation and safety. Under the assumption that the best arm w.r.t. its essential infimum is equivalent to the best arm w.r.t. its expectation, they show that their algorithm MIN, achieves the same regret as *UCB1*. Under the additional assumptions that the empirical minimum value for every arm converges exponentially fast towards its essential infimum and, with high probability over all arms, the empirical minimum values are exponentially close to their essential minimum, where the probability increases exponentially fast with the number of iterations, they show that *MIN might* outperform *UCB1*.

Zimin et al. [2014] considers a risk objective that evaluates the quality of an arm by some general function of the mean and the variance, generalizing our result from a linear to arbitrary functions. They present conditions under which learning is possible for continuous and discontinuous functions, proposing algorithms with log regret under both function settings. In the discontinuous case, they make the assumption that arms should not hit the discontinuity points of the risk measure. They also present examples for which no natural algorithm can achieve sublinear regret.

Yu and Nikolova [2013] consider the problem of minimizing the value-at-risk, the average value-at-risk, and the mean-variance risk over a single and multiple time periods, along with PAC accuracy guarantees given a finite number of reward samples. In particular, they study the complexity of estimating decision-theoretic risk in sequential decision-making as the number of arms increases and in the best arm identification setting. When considering single-period risk, they show that the arm with the least risk requires not many more samples than the arm with the highest expected reward. Under the multi-period setting, they present an algorithm for estimating the value-at-risk with comparable sample complexity under additional assumptions.

Tran-Thanh and Yu [2014] introduce the functional bandit problem to finite-horizon best-arm identification, where the objective is to find the arm that optimizes a known functional of the (unknown) arm distributions under a known time horizon. They propose the *Batch Elimination* algorithm, which combines functional estimation and arm elimination to achieve efficient performance guarantees.

Maillard [2013] study the standard expectation maximization objective instead of the risk objective introduced in this work. Similar to Audibert et al. [2009], they study the risk of the regret deviating from its expectation. In particular, they characterize risk according to the variability of the arm distributions and control this risk using a coherent measure of risk on the tail of the regret distribution.

Chapter 4

Online Learning with a Benchmark

Contents

1 Introduction 87

2 Preliminaries 91

3 Risk in Online Learning 99

4 Online Learning with a *flexible* Benchmark 105

5 Applications 113

6 Empirical Results 123

7 Conclusions 126

1 Introduction

In the previous chapter, we studied the influence of online estimation of a risk objective on decision-making in the stochastic multi-arm bandit setting. By setting a stochastic environment, it was possible to study the decision-making policy under efficient and unbiased estimates. The impact on performance could be directly attributed to the decision policy and its ability to evaluate the risk objective. We showed that the regret performance depends on the difficulty of discriminating the mean and variance gap between arms and that identifying the optimal arm according to a risk objective was challenging. The impact of identifying the gap was illustrated in detail through detailed sensitivity analysis. A sequential decision-making algorithm relying on an upper confidence bound and active decision policy, making several choices over the horizon, suffered a constant regret

in the worst-case, while a policy that makes a single decision relying on best arm identification, in an explicit exploration phase, resulted in relatively poor regret guarantees. Thus suggesting that estimating risk objectives, while simultaneously evaluating choices, adversely impacts sequential decision-making, and that managing risk objectives, under partial information, is hard. While results for stable distributions (stochastic environment) and partial observation were challenging, here we deepen our study to unstable distributions (adversarial environment) and full observation. Thus allowing us to study the impact of risk objectives on policies from an alternative perspective.

Here we study adversarial[1] full-information online learning. Accordingly, we make no statistical assumptions on the process and performance guarantees hold in for any possible sequence of observations. This means that results hold for *any* nonstationary, stationary or stochastic process. Variations on the shape of the decision set and loss functions give this setting its generality. Example problem settings include online convex optimization, sequential investment, prediction with expert advice and tracking the best expert, with applications in online portfolio selection, stock prediction, resource allocation, time series forecasting and data aggregation, among others (see e.g.,Cesa-Bianchi and Lugosi [2006] for an overview).

One setting that is often studied is *Learning with Expert Advice*, where algorithms maintain a set of *beliefs* over experts. A decision policy evaluates beliefs and the algorithm chooses which action to take at each round. Their aim is to perform close to the single best expert in hindsight, where the difference between the algorithm's choice and this best expert at each round is called the instantaneous regret. A positive result for this setting is when the cumulative per-step regret goes to zero or average per-step regret is constant as time goes to infinity,

[1]In particular, we assume an "oblivious" adversary. Note that this is not equal to a non-oblivious, adaptive, full or reactive adversary that chooses loss values according to the algorithm's actions on all previous rounds. This environmental setting assumes that the adversary is deterministic and has unlimited computational power, which implies that the adversary can compute an optimal policy for reacting to any possible sequence of actions chosen by the algorithm (please see e.g., Cesa-Bianchi and Lugosi [2006] for an overview). Throughout this chapter, we drop "oblivious" when referring to an "oblivious" adversary and make clear any references to an adaptive adversary.

and is referred to as a "no-regret" result. These robust regret guarantees are for any possible sequence, but it does not consider any notion of risk in the way the performance of experts is evaluated.

Two previous works study risk-aversion in this setting. Even-Dar et al. [2006] studies the average regret, while Warmuth and Kuzmin [2012] studies the cumulative regret. Each work studies risk-aversion from the Markowitz Mean–Variance perspective [Markowitz, 1952]. They show "no-regret" results under both regret settings in the case that the risk measure is measurable from the observed history of observations [Even-Dar et al., 2006] or fully revealed by the environment [Warmuth and Kuzmin, 2012], while [Even-Dar et al., 2006] proves a negative result in the case of partial observation of the risk measure. Thus demonstrating that the sequential decision-making policy requires a risk measure that is fully observable (measurable) or revealed by the environment to properly evaluate choices.

First, Even-Dar et al. [2006] study average regret (per-step) with a Markowitz [1952] Mean–Variance objective defined by the mean and standard deviation, where the regret is measured in terms of the Mean–Variance of the algorithm compared with the Mean–Variance of the best expert in hindsight[2]. They prove that no algorithm can achieve a "no-regret" result and that this is even true when the observation sequence is fully observed. No algorithm relying on any possible Mean-Variance objective can avoid a constant (average) regret. Their analysis reveals an unavoidable regret penalty caused by changing decisions between rounds. Recall, this switching penalty also exists in the bandit setting, studied in Chapter 3. Even-Dar et al. [2006] remove this penalty by introducing an alternative risk measure based on fully observable (and measurable) historical observations, which results in a positive result for the average regret. Second, Warmuth and Kuzmin [2012] study cumulative regret with a decision set defined over mixtures of experts and a Markowitz [1952] Mean-Variance objective defined by a loss and covariance matrix (Variance–Loss) revealed at each round by the environment (e.g., the

[2]Many alternative definitions of the Mean–Variance exist. The most common change is replacing the variance term by the standard deviation. Another modification is with regard to observations. The optimization expression is over mean values [Even-Dar et al., 2006] or per-round observations [Warmuth and Kuzmin, 2012].

learner must solve a Variance–Loss optimization problem at each round). Thus, as in Even-Dar et al. [2006], this setting also has access to an accurate measure of risk, achieving a positive result for the cumulative regret.

The Mean-Variance objective was selected in the previous chapter to study a particular characteristic of the decision-making process. Results in the online learning literature confirm that risk objectives require observable measures of risk, which are either revealed by the environment [Warmuth and Kuzmin, 2012] or observable from the data [Even-Dar et al., 2006]. By requiring a fully specified risk measure or limiting its impact to the past, the literature demonstrates that such risk measures are unrealistic in adversarial environments, where no actual distribution is defined. In practice, this is not always possible. Further, we may need to characterize risk with reference to some existing signal or algorithm. A natural choice of managing risk is to "hedge" risk to some benchmark. Unrelated to risk objectives in online learning, Even-Dar et al. [2008] introduce a novel regret analysis that measures performance based on simultaneous regret bounds to the best expert in hindsight and a fixed allocation over experts. This chapter extends the *fixed* benchmark of D-PROD to a flexible risk "hedging" structure in $(\mathcal{A}, \mathcal{B})$-PROD, accepting any fixed, changing or adaptive benchmark that can even learn. $(\mathcal{A}, \mathcal{B})$-PROD mixes over the decisions of a learning algorithm $\mathcal{A}$, with worst-case guarantees, and any benchmark $\mathcal{B}$ to achieve the best possible performance from either algorithm. This novel risk-aware structure results in a principled mechanism to "hedge" risk in any full-information sequential decision-making problem. Our method guarantees a constant regret w.r.t. any existing **benchmark** strategy together with small regret against the **best strategy** in hindsight. This is particularly useful in domains where the learning algorithm should be **safe** and **never worsen** the performance of an existing strategy.

This chapter studies the application of exploiting "easy" data, while dealing with sequences of "easy" and "hard" data sequences. The benchmark is set to an algorithm that focuses on exploiting "easy" data sequences. This problem recently received much attention in a variety of settings (see, e.g., [de Rooij et al., 2014] and [Grunwald et al., 2013]).

Input: Decision set $\mathcal{S}$, Rounds T, Function class $f \in \mathcal{F}$
For all $t = 1, 2, \ldots, T$**, repeat**

1. Simultaneously
   - Environment chooses $f_t \in \mathcal{F}$.
   - Learner chooses decision $\mathbf{x}_t \in \mathcal{S}$.
2. Environment reveals f_t.
3. Learner suffers loss $f_t(\mathbf{x}_t)$.
4. Learner updates beliefs.

end for

Figure 4.1: Online Learning Protocol

The structure of the chapter is as follows. First, the adversarial full-information setting is introduced in Section 2.1. An overview of risk-sensitive online learning is presented in Section 3. In Section 4, $(\mathcal{A}, \mathcal{B})$-Prod is introduced along with theoretical guarantees. Section 5 presents results on multiple problem settings in the "easy" and "hard" benchmark setting. Finally, empirical performance on standard loss sequences is presented in Section 6.

2 Preliminaries

We now formally introduce the online learning setting, with the interaction protocol described in Figure 4.1. The basic protocol is as follows. First, the environment chooses a loss function $f_t : \mathcal{S} \to [0, 1]$. Simultaneously, the learner chooses a decision $\mathbf{x}_t \in \mathcal{S}$, based on previous observations and possibly some external source of randomness. Next, the environment reveals f_t and the learner suffers loss $f_t(\mathbf{x}_t)$. Finally, the learner updates its beliefs.

2.1 Online Learning with Full Information

We consider the general class of online (sequential) decision-making problems following the protocol in Fig. 4.1, where the learner's objective is to minimize its

Input: Experts $\{1, \dots, K\}$, Decision set $\mathcal{S} = \Delta_K$, Rounds T, Function class $\mathcal{F} \in [0,1]^K$.
Initialize: $w_{i,1} = 1, \forall i$.
For all $t = 1, \dots, T$**, repeat**

1. Simultaneously
   - Environment chooses $\mathbf{l}_t \in \mathcal{F}$.
   - Learner chooses decision $\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathcal{S}} \mathbf{x}^\top \mathbf{w}_t$.
2. Environment reveals $\mathbf{l}_t$.
3. Learner suffers loss $f_t(\mathbf{x}_t) = \mathbf{x}_t^\top \mathbf{l}_t$.
4. Learner updates weights $\mathbf{w}_{t+1}$, where $w_{i,t+1} = w_{i,t} + l_{i,t}, \forall i$.

end for

Figure 4.2: Follow the Leader (FTL)

cumulative loss,

$$L_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t),$$

and achieve a cumulative loss close to the single best decision (expert[3]) in hindsight,

$$L_T^* = \arg\min_{\mathbf{x} \in S} \sum_{t=1}^{T} f_t(\mathbf{x}),$$

where performance is measured with regard to its cumulative *regret*,

$$\mathcal{R}_T = L_T - L_T^*,$$

That is, the difference between the cumulative loss of the algorithm L_T and that of the best single decision in hindsight L_T^*. Ultimately, the learner's objective is to achieve a sublinear cumulative regret, $\mathcal{R}_T = o(T)$. Note that throughout this chapter, we denote the regret of an algorithm $\mathcal{A}$ with respect to a sequence of decisions $\mathbf{x} = \{x_1, \dots, x_T\}$ by $\mathcal{R}(\mathcal{A}, \mathbf{x})$.

[3]Recall that decisions in the full information setting are referred to as *experts* as opposed to the bandit setting in Chapter 3, where they are referred to as *arms*.

2.2 Prediction with Expert Advice

We first consider the most basic online optimization problem of learning with expert advice. Here, $\mathcal{S}$ is the K-dimensional simplex $\Delta_K = \left\{\mathbf{x} \in \mathbb{R}_+^K : \sum_{i=1}^K x_i = 1\right\}$ and the loss functions are linear, that is, the loss of any decision $\mathbf{x} \in \Delta_K$ in round t is given as the inner product $f_t(x) = \mathbf{x}^\top \mathbf{l}_t$ and $\mathbf{l}_t \in [0,1]^K$ is the loss vector in round t. Accordingly, the family $\mathcal{F}$ of loss functions can be represented by the set $[0,1]^K$.

2.3 Weighted Majority Algorithms

This chapter studies WM algorithms that use a multiplicative update to iteratively maintain expert weights (for a review of the WM method, please see e.g., Arora et al. [2012]). Many algorithms are known to achieve the optimal regret guarantee of $\mathcal{O}(\sqrt{T \log K})$ in this setting, including HEDGE (Vovk [1990], Littlestone and Warmuth [1994], Freund and Schapire [1997]) and Follow the Perturbed Leader *FPL* introduced by Hannan [1957] and later rediscovered by Kalai and Vempala [2005]. When the learning rate is appropriately tuned, WM algorithms guarantee worst–case regret of order $\mathcal{O}(\sqrt{T})$ in this setting [Cesa-Bianchi and Lugosi, 2006], where results hold for any (possibly adversarial) assignment of the loss sequence. Thus, these algorithms are guaranteed to achieve "no–regret" performance even in the worst–case. Furthermore, there exist sequences of loss functions where the learner suffers $\Omega(\sqrt{T})$ regret no matter what algorithm is used, so these guarantees are "unimprovable" in the worst-case.

One simple WM algorithm is Follow the Leader (*FTL*) (see, e.g., Cesa-Bianchi and Lugosi [2006] and Figure 4.2), which chooses the decision that minimizes the observed sequence of losses. When assuming a benign adversary or i.i.d. loss vectors in the expert setting, *FTL* guarantees $\mathcal{O}(\log T)$ regret. This guarantee also holds in several other settings, such as online convex optimization (see e.g.,Hazan et al. [2007a]), where the assumption is that all loss functions are strongly convex. Unfortunately, *FTL* only learns on easy observations, such as i.i.d., and fails to learn in the worst-case, suffering $\Omega(T)$ regret.

Input: Learning rate $\eta > 0$, Experts $\{1, \ldots, K\}$, Decision set $\mathcal{S} = \Delta_K$, Rounds T, Function class $\mathcal{F} \in [0,1]^K$.
Initialize: $w_{i,1} = 1, \forall i$.
For all $t = 1, \ldots, T$**, repeat**

1. Simultaneously
   - Environment chooses $\mathbf{l}_t \in \mathcal{F}$.
   - Learner chooses decision $\mathbf{x}_t \in \mathcal{S}$, where $x_{i,t} = \frac{w_{i,t}}{\sum_{i=1}^{K} w_{i,t}}, \forall i$.
2. Environment reveals $\mathbf{l}_t$.
3. Learner suffers loss $f_t(\mathbf{x}_t) = \mathbf{x}_t^\top \mathbf{l}_t$.
4. Learner updates weights $\mathbf{w}_{t+1}$, where $w_{i,t+1} = w_{i,t} \exp(-\eta l_{i,t}), \forall i$.

end for

Figure 4.3: HEDGE

HEDGE, otherwise known as the aggregating algorithm or exponentially weighted forecaster (see e.g., Figure 4.3), reduces to *FTL* when the learning rate is set to $\eta = \infty$. Worst–case regret guarantees are reported in Theorem 6 along with a proof.

Theorem 6. *[Cesa-Bianchi and Lugosi, 2006, Chapter 2] For any T, $\eta > 0$ and learning rate $\eta = \sqrt{\frac{8 \log K}{T}}$, the regret upper bound of* HEDGE *satisfies,*

$$\mathcal{R}_T(Hedge, \mathbf{x}) \leq \sqrt{\frac{T}{2} \log K},$$

for any $\mathbf{x} \in \mathcal{S}$.

Proof. [Cesa-Bianchi and Lugosi, 2006, Chapter 2]
Set finite time horizon T, experts K, weights $W_t = \sum_{i=1}^{K} w_{i,t}$, loss $l_{i,t} \in [0,1]$ and update $w_{i,t+1} = \exp(-\eta l_{i,t})$. Initialize $\forall i w_{i,1} = 1$, where $W_1 = K$.

Then, $\forall i \in \{1, \ldots, K\}$,

$$\log \frac{W_T}{W_1} = \log(W_T) - \log W_1 \tag{4.1}$$

$$= \log \left(\sum_{k=1}^{K} w_{i,T} \right) - \log K \tag{4.2}$$

$$\geq \log \left(\max_{i=1,\ldots,K} \exp(-\eta L_{i,T}) \right) - \log K \tag{4.3}$$

$$= -\eta \min_{i=1,\ldots,K} L_{i,T} - \log K \tag{4.4}$$

$$= -\eta L_{i^*,T} - \log K. \qquad \left(i^* = \min_{i=1,\ldots,K} L_{i,T} \right) \tag{4.5}$$

Furthermore, for $t = 1, 2, \ldots, T$,

$$\log \frac{W_{t+1}}{W_t} = \log \left(\sum_i \frac{w_{i,t} \exp\left(-\eta l_{i,t}\right)}{W_t} \right) \qquad \textbf{(Update Rule)} \tag{4.6}$$

$$= \log \left(\sum_i \frac{w_{i,t}}{W_t} \exp\left(-\eta l_{i,t}\right) \right) \tag{4.7}$$

$$\leq \frac{\eta^2}{8} - \eta \sum_i^K p_{i,t} l_{i,t}, \qquad \textbf{(Chernoff–Hoeffding Bound)} \tag{4.8}$$

where $p_{i,t}$ is the probability of expert i at time t. Summing up for all t and combining the above inequalities, we get,

$$-\eta L_{i^*,T} - \log K \leq \sum_{t=1}^{T} \left(\frac{\eta^2}{8} - \eta \sum_i^K p_{i,t} l_{i,t} \right) \tag{4.9}$$

$$\eta \sum_{t=1}^{T} \sum_i^K p_{i,t} l_{i,t} - \eta L_{i^*,T} \leq \frac{T\eta^2}{8} + \log K, \tag{4.10}$$

where we divide by η to get the regret,

$$\mathcal{R}_T \leq \frac{T\eta}{8} + \frac{\log K}{\eta}, \tag{4.11}$$

Input: Learning rate $\eta \in \left(0, \frac{1}{2}\right]$, Experts $\{1, \ldots, K\}$, Decision set $\mathcal{S} = \Delta_K$, Rounds T, Function class $\mathcal{F} \in [0,1]^K$.
Initialize: $w_{i,1} = 1, \forall i$.
For all $t = 1, \ldots, T$**, repeat**

1. Simultaneously
   - Environment chooses $\mathbf{l}_t \in \mathcal{F}$.
   - Learner chooses decision $\mathbf{x}_t \in \mathcal{S}$, where $x_{i,t} = \frac{w_{i,t}}{\sum_{i=1}^{K} w_{i,t}}, \forall i$.
2. Environment reveals $\mathbf{l}_t$.
3. Learner suffers loss $f_t(\mathbf{x}_t) = \mathbf{x}_t^\top \mathbf{l}_t$.
4. Learner updates weights $\mathbf{w}_{t+1}$, where $w_{i,t+1} = w_{i,t}(1 - \eta l_{i,t}), \forall i$.

end for

Figure 4.4: PROD

and finally, solve with optimized $\eta = \sqrt{\frac{8 \log K}{T}}$, to get the result,

$$\mathcal{R}_T(Hedge, \mathbf{x}) \leq \sqrt{\frac{T}{2} \log K}, \tag{4.12}$$

for any $\mathbf{x} \in \mathcal{S}$. □

PROD [Cesa-Bianchi et al., 2007] is another WM algorithm, also known as the multilinear forecaster in Cesa-Bianchi and Lugosi [2006], which achieves the second-order bounds reported in Theorem 7 to any individual expert. One difference between PROD and HEDGE (that will be elaborated in the discussion towards the end of the chapter) is the difference in updates, where the weight of an expert k at time t is no longer updated exponentially as in $w_{k,t+1} = w_{k,t} \exp(-\eta l_{k,t})$, but linearly through $w_{k,t+1} = w_{k,t}(1 - \eta l_{k,t})$ (for details see e.g., Cesa-Bianchi et al. [2007]).

Theorem 7. *[Cesa-Bianchi and Lugosi, 2006, Chapter 2] For any T and learning rate $0 \leq \eta \leq \frac{1}{2}$, Prod satisfies the following second-order regret bound,*

$$\mathcal{R}_T(Prod, i) \leq \eta T + \frac{\log K}{\eta}. \tag{4.13}$$

for any $\mathbf{x} \in \mathcal{S}$ and the following regret bound with optimized learning rate $\eta =$

$\sqrt{\frac{\log K}{T}}$,

$$\mathcal{R}_T(Prod, \mathbf{x}) \leq 2\sqrt{T \log K}. \tag{4.14}$$

Proof. Set finite time horizon T, experts K, weights $W_t = \sum_{i=1}^{K} w_{i,t}$, loss $\ell_{i,t} \in [0,1]$ and update $w_{i,t+1} = (1 - \eta l_{i,t})$. Initialize $w_{i,1} = 1, \forall i$, where $W_1 = K$. Then, $\forall i \in \{1, \ldots, K\}$,

$$\begin{aligned}
\log \frac{W_{T+1}}{W_1} &= \log W_{T+1} - \log K && (4.15)\\
&\geq \log \prod_{t=1}^{T} (1 - \eta l_{i,t}) - \log K && (4.16)\\
&= \sum_{t=1}^{T} \log(1 - \eta l_{i,t}) - \log K && (4.17)\\
&\geq -\eta L_{i,t} - \eta^2 \sum_{t=1}^{T} l_{i,t}^2 - \log K, && (4.18)
\end{aligned}$$

where we used the inequality $\log(1 - X) \geq -X - X^2$, for all $0 \leq X \leq \frac{1}{2}$. Furthermore, for any $t = 1, 2, \ldots, T$, we have,

$$\begin{aligned}
\log \frac{W_{t+1}}{W_t} &= \log \left(\sum_{i}^{K} \frac{w_{i,t}}{W_t} (1 - \eta l_{i,t}) \right) && (4.19)\\
&= \log \left(\sum_{i}^{K} p_{i,t} (1 - \eta l_{i,t}) \right) && (4.20)\\
&= \log \left(1 - \eta \sum_{i}^{K} p_{i,t} l_{i,t} \right) && (4.21)\\
&\leq -\eta \sum_{i}^{K} p_{i,t} l_{i,t}, && (4.22)
\end{aligned}$$

where we use the inequality, $\log(1 - X) \leq -X$. Summing up for all t and combining

the above inequalities, we get,

$$-\eta L_{i,t} - \eta^2 \sum_{t=1}^{T} l_{i,t}^2 - \log K \leq -\eta \sum_{t=1}^{T} \sum_{i}^{K} p_{i,t} l_{i,t} \tag{4.23}$$

$$\eta \sum_{t=1}^{T} \sum_{i}^{K} p_{i,t} l_{i,t} - \eta L_{i,t} \leq \eta^2 \sum_{t=1}^{T} l_{i,t}^2 + \log K, \tag{4.24}$$

where we divide by η,

$$L_T - L_{i,T} \leq \eta \sum_{t=1}^{T} l_{i,t}^2 + \frac{\log K}{\eta}, \tag{4.25}$$

and upper bound $\sum_{t=1}^{T} l_{i,t}^2 \leq 1$,

$$L_T - L_{i,T} \leq \eta T + \frac{\log K}{\eta}, \tag{4.26}$$

setting $L_{i,t}$ to the best expert L_t^*,

$$L_T - L_T^* \leq \eta T + \frac{\log K}{\eta}, \tag{4.27}$$

to get the second-order regret bound,

$$\mathcal{R}_T(Prod, k) \leq \eta T + \frac{\log K}{\eta}. \tag{4.28}$$

and finally, solve with optimized $\eta = \sqrt{\frac{\log K}{T}}$, to get the result,

$$\mathcal{R}_T(Prod, \mathbf{x}) \leq 2\sqrt{T \log K}, \tag{4.29}$$

for any $\mathbf{x} \in \mathcal{S}$. □

3 Risk in Online Learning

3.1 Risk Sensitive Online Learning

Even-Dar et al. [2006] study risk–averse online learning in signed games[4]. The setting is the same as in prediction with expert advice, except that the choice at each round is a single expert $I_t \in \{1, \ldots, K\}$. Further, the regret is no longer measured w.r.t. the mean loss[5] but it rather focuses on the mean and variance of the losses incurred by the algorithm. More precisely, each expert k is evaluated according to

$$\mathrm{MD}_T(k) = \mu_{k,T} + \sigma_{k,T}, \tag{4.30}$$

where $\mu_{k,T} = \frac{1}{T}\sum_{t=1}^{T} l_{k,t}$ and $\sigma_{k,T} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(l_{k,t} - \mu_{k,t})^2}$ are computed from the instantaneous losses $l_{k,t}$. Similarly, the performance of an algorithm $\mathcal{A}$, which selects an expert I_t at each step t is evaluated according to

$$\mathrm{MD}_T(\mathcal{A}) = \mu_{\mathcal{A},T} + \sigma_{\mathcal{A},T}, \tag{4.31}$$

where $\mu_{\mathcal{A},T} = \frac{1}{T}\sum_{t=1}^{T} l_{I_t,t}$ and $\sigma_{\mathcal{A},T} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(l_{I_t,t} - \mu_{\mathcal{A},t})^2}$. Finally, the objective of an algorithm is to minimize the (per-step) regret

$$r_T(\mathcal{A}, k) = \mathrm{MD}_T(\mathcal{A}) - \min_{k \in K} \mathrm{MD}_T(k),$$

and in particular to obtain a regret which vanish to zero as T increases. The first result derived by Even-Dar et al. [2006] is that unfortunately there exists no algorithm $\mathcal{A}$ with decreasing regret, as stated in the following theorem.

Theorem 8. *Let $\rho \geq 0$ be a constant. Then, the regret of any online algorithm with respect to the metric $\mu + \rho\sigma$ is lower bounded by some positive constant C*

[4] Note that the original objective, $\mu_{i,T} - \sigma_{i,T}$ in Even-Dar et al. [2006] is for rewards in $[-1, \infty]$, with the aim of maximization. Here we assume losses in $[0, 1]$ and the aim of minimization, so we change the subtraction to an addition.

[5] Notice by dividing the *cumulative* regret $\mathcal{R}_T$ by T, we obtain the so-called *per-step* regret $r_T = \frac{R_T}{T}$ which compares the average loss of the algorithm to the average loss of the best expert in hindsight.

Input: Weighted Majority Algorithm $\mathcal{A}$, Learning rate η, Rewards $l_{i,t} \in [0,1]$, Experts $\{1,\dots,K\}$, Decision set $\mathcal{S} = \Delta_K$, Rounds T, Function class $\mathcal{F} \in [0,1]^K$.
Initialize: $I_t = \mathcal{A}(\mathcal{U}([1,K]))$.
For all $t = 1,\dots,T$**, repeat**

1. Simultaneously
   - Environment chooses $\mathbf{l}_t$.
   - Learner chooses expert $I_t = \mathcal{A}$ (i.e., the expert suggested by algorithm $\mathcal{A}$)
2. Environment reveals $\mathbf{l}_t$.
3. Learner computes pseudo-loss $\tilde{l}_{k,t} = l_{k,t} - \xi_{k,t}$
4. Learner updates algorithm $\mathcal{A}$ with pseudo-losses $\{\tilde{l}_{k,t}\}_{k=1}^K$

end for

Figure 4.5: Mean–Deviation [Even-Dar et al., 2006]

that depends on the risk aversion parameter ρ, that is:

$$r_T(\mathcal{A}, k) = \mathrm{MD}_T(\mathcal{A}) - \mathrm{MD}_T(k) \geq C.$$

This represents a strong negative result on the possibility to achieve a risk-averse objective in online learning. They conjecture that this risk sensitive objective introduces a "switching cost" not present in the standard setting, where "no-regret" algorithms are possible because the learner is not directly penalized for switching between experts. According to their analysis, it is impossible to determine the best expert in the case of unrealized variance.

A clear support to this conjecture is provided by the positive results that can be achieved by slightly modifying the definition of variance. In particular, they introduce an alternative risk measure restricted to the observable history of losses, defined as,

$$\begin{aligned} P_{k,T} &= \sum_{t=2}^{T} \left(l_{k,t} - \sum_{s=0}^{d-1} \frac{l_{k,t-s}}{d} \right)^2 \\ &= \sum_{t=2}^{T} \xi_{k,t}, \end{aligned} \tag{4.32}$$

where the variance is now computed by considering only a fixed-size window mean

Input: Learning rate η, Experts $\{1, \dots, K\}$, Decision set $\mathcal{S} = \Delta_K$, Rounds T, Function class $\mathcal{F} \in [0,1]^K$.
Initialize: $w_{i,1} = 1, \forall i$.
For all $t = 1, \dots, T$**, repeat**

1. Simultaneously
   - Environment chooses $\mathbf{l}_t$ and $\mathbf{C}_t$.
   - Learner chooses decision $\mathbf{x}_t \in \mathcal{S}$, where $x_{i,t} = \frac{w_{i,t}}{\sum_{i=1}^K w_{i,t}}, \forall i$.
2. Environment reveals $\mathbf{l}_t$ and $\mathbf{C}_t$.
3. Learner suffers loss $VL_t(\mathbf{x}_t) = \rho \mathbf{x}_t^\top \mathbf{l}_t + \mathbf{x}_t^\top \mathbf{C}_t \mathbf{x}_t$.
4. Learner updates weights $\mathbf{w}_{t+1}$, where $w_{i,t+1} = w_{i,t} \exp(-\eta(\rho l_{i,t} + (\mathbf{C}_t \mathbf{x}_t)_i)), \forall i$.

end for

Figure 4.6: Variance–Loss [Warmuth and Kuzmin, 2012]

of size d and $\xi_{k,t}$ is the risk of k at time t. Given the definition of $P_T(k)$, $\mathrm{MD}_T(k)$ is then replaced by

$$\mathrm{MP}_T(k) = \mu_{k,T} + \frac{P_{k,T}}{T}, \tag{4.33}$$

and the regret is redefined accordingly. Even-Dar et al. [2006] show that any no-regret algorithm can then be easily employed to solve this problem. In fact, it is enough the create a pseudo-loss

$$\tilde{l}_{k,t} = l_{k,t} - \xi_{k,t}, \tag{4.34}$$

and use it as input for an algorithm $\mathcal{A}$ as illustrated in Figure 4.5 to achieve a zero-regret for the regret minimization w.r.t. the objective MP_T.

Theorem 9. *Let $\mathcal{A}$ be a WM algorithm with updates based on the pseudo-loss in Eq. 4.34 and $\eta = \sqrt{\frac{\log(K)}{T}}$. Then for large enough T, any losses $l_{k,t} \in [0,1]$, the per-step regret upper bound satisfies for any expert k*

$$r_T(\mathcal{A}, k) \leq \mathcal{O}\left(d\sqrt{\frac{\log K}{T-d}}\right). \tag{4.35}$$

3.2 Online Variance–Loss Minimization

Warmuth and Kuzmin [2012] use a variance–loss objective to update HEDGE in an online convex optimization setting (see e.g., Zinkevich [2003]). This is simply the Mean–Variance objective for instantaneous losses instead of the mean. The algorithm is formally presented in Figure 4.6. At each round, the environment reveals a loss $\mathbf{l}_t$ and the covariance matrix $\mathbf{C}_t$ and the learner must minimize the variance–loss tradeoff,

$$VL_t(\mathbf{x}_t) = \rho \mathbf{x}_t^\top \mathbf{l}_t + \mathbf{x}_t^\top \mathbf{C}_t \mathbf{x}_t$$

according to a given risk aversion parameter $0 \leq \rho \leq \infty$. Though they note that the covariance in this setting can be estimated, it is unrealistic in practice to assume the availability of the actual covariance matrix at the end of each step. The learner's cumulative loss is defined as,

$$VL_T = \sum_{t=1}^{T} VL_t(\mathbf{x}_t),$$

and the aim of the learner is to achieve a performance close to the best single mixture over experts in hindsight,

$$VL_T^* = \arg\min_{x \in \Delta_K} \left(\rho \mathbf{x} \sum_{t=1}^{T} \mathbf{l}_t + \mathbf{x}^\top \left(\sum_{t=1}^{T} \mathbf{C}_t \right) \mathbf{x} \right),$$

and to minimize the cumulative regret,

$$\mathcal{R}_T^{VL}(\mathcal{A}, x) = VL_T - VL_T^*.$$

The regret bounds match those of Zinkevich [2003] and are presented in Theorem 10.

Theorem 10. *[Warmuth and Kuzmin, 2012] Let $0 \leq \rho \leq \infty$ be the risk aversion parameter, $\mathbf{C}_1, \ldots, \mathbf{C}_T$ be an arbitrary sequence of covariance matrices such that $\max_{i,j} |[\mathbf{C}_t]_{i,j}| \leq \frac{r}{2}$ and $\mathbf{l}_1, \ldots, \mathbf{l}_T$ be an arbitrary sequence of loss vectors such that $l_{i,t} \in [0,1]$. Additionally assume an upper bound on the losses $\mathbf{L} \geq \rho \mathbf{x} \sum_{t=1}^{T} \mathbf{l}_t +$*

$\mathbf{x}^\top \left(\sum_{t=1}^{T} \mathbf{C}_t\right) \mathbf{x}$, *for all* $\mathbf{x} \in \Delta_K$. *Then* HEDGE *on the* K*-dimensional probability simplex with uniform start vector* $\boldsymbol{w}_0 = \left(\frac{1}{K}, \ldots, \frac{1}{K}\right)$ *and learning rate*

$$\eta = \frac{2\sqrt{\frac{\log K}{Q\mathbf{L}}}\,(r - 2\rho)}{r + \sqrt{\frac{\log K}{Q\mathbf{L}}}\,(r + \rho)^2},$$

where $Q = \frac{(r+\rho)^2}{r}$, *has the following (cumulative) regret,*

$$\mathcal{R}_T(Hedge, \mathbf{x}) \leq \frac{2}{R}\sqrt{Q\mathbf{L}\log K} + \frac{Q}{R}\log K + \mathbf{L}P.$$

where $R = \frac{(r-2\rho)^2}{r(r+2\rho)}$ *and* $P = \left(\frac{2\rho(3r-2\rho)}{(r-2\rho)^2}\right)$.

Note that it is quite unrealistic to assume that the environment reveals the true covariance function or that it can effectively be estimated at each round by any estimator to avoid impacting the regret. This assumption is very strong on the environment. In many applications, especially in finance, accurately estimating the covariance matrix is very hard. This is the case in all types of observation sequences, including simple i.i.d. sequences, over a single observation. By assuming the environment reveals the actual covariance matrix at each time step, this setting avoids the complex estimation problem of learning the covariance structure between experts within the adversarial setting. Note that unlike the simple i.i.d. setting that would assume a fixed covariance matrix over all rounds, where each sample at each round from each expert can be used to improve covariance matrix estimates, the adversarial setting assumes no fixed covariance over rounds. In application areas such as finance, where the covariance plays a significant role in risk estimation, the divergence between the empirical and actual covariance matrix has been studied and results show the estimation error can be substantial Vershynin [2012]. Further, misspecification of the covariance matrix can result in highly inaccurate allocations Ledoit and Wolf [2004].

3.3 Risk to the *Best* versus Risk to the *Average*

Even-Dar et al. [2008] introduce a bicriteria interpretation of the regret that ex-

Input: Losses $l_{i,t} \in [0,1]$, Experts $\{1,\ldots,K\}$, Expert D, Decision set $\mathcal{S} = \Delta_K$, Rounds T, Learning rate $\eta = \sqrt{\frac{\log K}{T}}$, Initial weights $\mu_i = \frac{\eta}{K}, \forall i \in \{1,\ldots,K\}$, $\mu_0 = 1-\eta$, Function class $\mathcal{F} \in [0,1]^K$.
Initialize: $w_{i,1} = \mu_i, \forall i \in \{0,\ldots,K\}$.
For all $t = 1,\ldots,T$**, repeat**

1. Simultaneously
   - Environment chooses $\mathbf{l}_t \in \mathcal{F}$.
   - Learner chooses decision $\mathbf{x}_t \in \mathcal{S}$, where
   $$x_{i,t} = \frac{w_{i,t}}{\sum_{i=0}^{K} w_{i,t}},$$
   for $i \in \{0,\ldots,K\}$.
2. Environment reveals $\mathbf{l}_t$.
3. Learner gains $f_t(\mathbf{l}_t) = \mathbf{x}_t^\top \mathbf{l}_t$.
4. Learner updates weights $\mathbf{w}_{t+1}$, where,
$$w_{i,t+1} = w_{i,t}(1 - \eta(l_{i,t} - l_{0,t})), \forall i \in \{1,\ldots,K\}.$$

end for

Figure 4.7: D-Prod [Even-Dar et al., 2008]

tends the standard regret of *performing well against the best expert* to include *performance against any (given) fixed distribution*. The learning objective is to perform within a constant of a given (fixed) benchmark, while also achieving a loss close to the best expert in hindsight. Their theoretical results show that difference algorithms (such as Weighted Majority/Exponential Weights, Follow the Perturbed Leader, and Prod) that simply select experts based on the difference in their cumulative loss, where the individual losses are bounded in $[0,1]$, achieve $\mathcal{O}(\sqrt{T})$ (cumulative) regret to the best expert over a sequence of T rounds, while, in the worst case, suffering $\Omega(\sqrt{T})$ regret to any (fixed) allocation over experts. They then note that the product of these regrets is $\Omega(T)$ in the worst case and reveal that this performance bottleneck can only be overcome by "restarts", where weights are reset to the uniform allocation, or favoring weight updates for experts that show improved performance. This result is reported in Theorem 11.

Theorem 11. *Let* $\mathbf{L} \leq T$ *be an upper bound on the cumulative loss for any expert*

and D be a fixed uniform average allocation over experts. For any φ such that $0 \leq \varphi \leq \frac{1}{2}$, set the Exponential Weights algorithm $EW = EW(\eta)$, with learning rate $\eta = \mathbf{L}^{-(\frac{1}{2}+\varphi)}$. Then the bicriteria regret bound satisfies

$$\mathcal{R}_T(EW, \mathbf{x}) \leq \mathbf{L}^{\frac{1}{2}+\varphi}(1 + \log K),$$

for any $\mathbf{x} \in \mathcal{S}$ and

$$\mathcal{R}_T(EW, D) \leq \mathbf{L}^{\frac{1}{2}-\varphi}.$$

They resolve this limitation by exploiting the second-order regret bounds to a specific expert in PROD, where a special "zero" expert is set to a fixed (given) allocation over experts is used as a Benchmark to difference individual expert losses in the PROD loss update in D-PROD (see Figure 4.7). This Benchmark is not used in the expert weight updates and results in a cancellation of the first term in the second-order bound (this will be explored in detail in Section 4.2). The simultaneous regret bounds for D-PROD are reported in Theorem 12.

Theorem 12. *Let $\eta = \sqrt{\left(\frac{\log K}{T}\right)}, \mu_0 = 1 - \eta$, and $\mu_i = \frac{\eta}{K}$ for $i \in \{1, \ldots, K\}$. Then the bicriteria regret bound satisfies*

$$\mathcal{R}_T(D\text{-PROD}, \mathbf{x}) = \mathcal{O}\left(\sqrt{T \log K} + \sqrt{\frac{T}{\log K}} \log T\right),$$

for any $\mathbf{x} \in \mathcal{S}$ and

$$\mathcal{R}_T(D\text{-PROD}, D) = \mathcal{O}(1),$$

against any fixed allocation over experts D.

4 Online Learning with a *flexible* Benchmark

This chapter introduces $(\mathcal{A}, \mathcal{B})$-PROD by modifying the structure of D-PROD to support a more general notion of benchmark that allows fixed, changing or adaptive strategy that can even learn. This endows a flexible interpretation that has many practical advantages. Learning algorithms with order-optimal regret bounds

are constructed by extending D-PROD, *while also guaranteeing a cumulative loss within a constant factor of some pre-defined strategy* referred to as a benchmark. We stress that this property is much stronger than simply guaranteeing $\mathcal{O}(1)$ regret with respect to some (given) fixed distribution D, as in [Even-Dar et al., 2008], since comparisons can now be made to *any fixed strategy that is allowed to learn and adapt to the problem.* More specifically, D-PROD is constrained to a (given) fixed benchmark mixture over experts, intrinsically defined by the experts setting. We extend this in generality to any problem setting by adding the flexibility of an adaptive algorithm that accepts a benchmark strategy that is allowed to learn, while also exploiting any advantage in mixing predictions with benchmark alternatives. Now that a brief review of previous works is complete, we move to a more formal introduction of our contribution.

4.1 $(\mathcal{A}, \mathcal{B})$-PROD

Let $\mathcal{A}$ and $\mathcal{B}$ be two online learning algorithms that map observation histories to decisions in a possibly randomized fashion. For a formal definition, we fix a time index $t \in [T] = \{1, 2, \ldots, T\}$ and define the observation history (or in short, the history) at the end of round $t-1$ as $\mathcal{H}_{t-1} = (f_1, \ldots, f_{t-1})$, where f_t takes values in $[0,1]$ from the function class $\mathcal{F}$. $\mathcal{H}_0$ is defined as the empty set, $\emptyset$. Furthermore, we define the random variables U_t and V_t, drawn from the standard uniform distribution, independently of $\mathcal{H}_{t-1}$ and each other. The learning algorithms $\mathcal{A}$ and $\mathcal{B}$ are formally defined as mappings from $\mathcal{F}^* \times [0,1]$ to $\mathcal{S}$ with their respective decisions given as

$$\mathbf{a}_t \stackrel{\text{def}}{=} \mathcal{A}(\mathcal{H}_{t-1}, U_t) \qquad \text{and} \qquad \mathbf{b}_t \stackrel{\text{def}}{=} \mathcal{B}(\mathcal{H}_{t-1}, V_t).$$

Finally, we define a *hedging strategy* $\mathcal{C}$ that produces a decision $\mathbf{x}_t$ based on the history of decisions proposed by $\mathcal{A}$ and $\mathcal{B}$, with the possible help of some external randomness represented by the uniform random variable W_t as

$$\mathbf{x}_t = \mathcal{C}\left(\mathbf{a}_t, \mathbf{b}_t, \mathcal{H}^*_{t-1}, W_t\right).$$

Here, $\mathcal{H}^*_{t-1}$ is the simplified history consisting of $\big(f_1(\mathbf{a}_1), f_1(\mathbf{b}_1), \ldots, f_{t-1}(\mathbf{a}_{t-1}), f_{t-1}(\mathbf{b}_{t-1})\big)$ and $\mathcal{C}$ bases its decisions only on the past losses incurred by $\mathcal{A}$ and $\mathcal{B}$ without using any further information on the loss functions. The total expected loss of $\mathcal{C}$ is defined as $\widehat{L}_T(\mathcal{C}) = \mathbb{E}[\sum_{t=1}^T f_t(\mathbf{x}_t)]$, where the expectation integrates over the possible realizations of the internal randomization of $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$. The total expected losses of $\mathcal{A}$, $\mathcal{B}$ and any fixed decision $\mathbf{x} \in \mathcal{S}$ are similarly defined.

Our goal is to define a hedging strategy with low regret against a benchmark strategy $\mathcal{B}$, while also enjoying near-optimal guarantees on the worst–case regret against the best decision in hindsight. The (expected) regret of $\mathcal{C}$ against any fixed decision $\mathbf{x} \in \mathcal{S}$ and against the benchmark, are defined as

$$\mathcal{R}_T(\mathcal{C}, \mathbf{x}) = \mathbb{E}\left[\sum_{t=1}^T \big(f_t(\mathbf{x}_t) - f_t(\mathbf{x})\big)\right], \tag{4.36}$$

$$\mathcal{R}_T(\mathcal{C}, \mathcal{B}) = \mathbb{E}\left[\sum_{t=1}^T \big(f_t(\mathbf{x}_t) - f_t(\mathbf{b}_t)\big)\right]. \tag{4.37}$$

Our hedging algorithm $(\mathcal{A}, \mathcal{B})$-Prod (shown in Figure 4.8) is based on the observation that an adaptive benchmark that is allowed to learn can be used in place of the fixed distribution D in the definition of the benchmark. $(\mathcal{A}, \mathcal{B})$-Prod maintains two weights, balancing the advice from a learning algorithm $\mathcal{A}$ and an adaptive benchmark $\mathcal{B}$. The benchmark weight is defined as $\mathbf{w}_{\mathcal{B},1} \in (0, 1)$ and is kept unchanged during the entire learning process. The initial weight assigned to $\mathcal{A}$ is $\mathbf{w}_{\mathcal{A},1} = 1 - \mathbf{w}_{\mathcal{B},1}$, and in the remaining rounds $t = 2, 3, \ldots, T$ is updated as

$$\mathbf{w}_{\mathcal{A},t+1} = \mathbf{w}_{\mathcal{A},1} \prod_{s=1}^{t} \Big(1 - \eta\big(f_s(\mathbf{a}_s) - f_s(\mathbf{b}_s)\big)\Big),$$

where the difference between the losses of $\mathcal{A}$ and $\mathcal{B}$ is used. Output $\mathbf{x}_t$ is set to $\mathbf{a}_t$ with probability $\mathbf{s}_t = \frac{\mathbf{w}_{\mathcal{A},t}}{\mathbf{w}_{\mathcal{A},t} + \mathbf{w}_{\mathcal{B},1}}$, otherwise it is set to $\mathbf{b}_t$[6]. The following theorem states the performance guarantees for $(\mathcal{A}, \mathcal{B})$-Prod.

[6]For convex decision sets $\mathcal{S}_K$ and loss families $\mathcal{F}$, one can directly set $\mathbf{x}_t = \mathbf{s}_t\mathbf{a}_t + (1 - \mathbf{s}_t)\mathbf{b}_t$ at no expense.

Input: Experts $\{1,\dots,K\}$, Decision set $\mathcal{S}$, Learning rate $\eta \in \left(0, \frac{1}{2}\right]$, Weights $w_{\mathcal{B},1} = (0,1), w_{\mathcal{A},1} = 1 - w_{\mathcal{B},1}$, Algorithms $\mathcal{A}$ and $\mathcal{B}$, Rounds T, Function class $\mathcal{F}$
Initialize: $\mathbf{a}_1 = \mathcal{A}(\emptyset, U_1)$ and $\mathbf{b}_1 = \mathcal{B}(\emptyset, V_1)$
For all $t = 1, 2, \dots, T$**, repeat**

1. Let $\mathbf{s}_t = \frac{\mathbf{w}_{\mathcal{A},t}}{\mathbf{w}_{\mathcal{A},t}+\mathbf{w}_{\mathcal{B},1}}$.
2. Simultaneously
   - Environment chooses $f_t \in \mathcal{F}$.
   - Learner predicts $\mathbf{x}_t = \begin{cases} \mathbf{a}_t & \text{with probability } \mathbf{s}_t, \\ \mathbf{b}_t & \text{with probability } 1 - \mathbf{s}_t. \end{cases}$
3. Environment reveals f_t.
4. Learner suffers loss $f_t(\mathbf{x}_t)$.
5. Learner draws uniform random variables U_t and V_t.
6. Learner observes $\mathbf{a}_t = \mathcal{A}(\{f_s\}_{s=1}^t, U_t)$ and $\mathbf{b}_t = \mathcal{B}(\{f_s\}_{s=1}^t, V_t)$.
7. Learner updates $\delta_t = f_t(\mathbf{a}_t) - f_t(\mathbf{b}_t)$.
8. Learner updates $\mathbf{w}_{\mathcal{A},t+1} = \mathbf{w}_{\mathcal{A},t}\,(1 - \eta\delta_t)$.

end for

Figure 4.8: $(\mathcal{A}, \mathcal{B})$-Prod

Theorem 13 (cf. Lemma 1 in Even-Dar et al. [2008])**.** *For any assignment of the loss sequence, the total expected loss of* $(\mathcal{A}, \mathcal{B})$-Prod *initialized with weights* $\mathbf{w}_{\mathcal{B},1} \in (0,1)$ *and* $\mathbf{w}_{\mathcal{B},1} = 1 - \mathbf{w}_{\mathcal{A},1}$ *simultaneously satisfies*

$$\widehat{L}_T((\mathcal{A},\mathcal{B})\text{-Prod}) \le \widehat{L}_T(\mathcal{A}) + \eta \sum_{t=1}^{T} \big(f_t(\mathbf{b}_t) - f_t(\mathbf{a}_t)\big)^2 - \frac{\log \mathbf{w}_{\mathcal{A},1}}{\eta}$$

and

$$\widehat{L}_T((\mathcal{A},\mathcal{B})\text{-Prod}) \le \widehat{L}_T(\mathcal{B}) - \frac{\log \mathbf{w}_{\mathcal{B},1}}{\eta}.$$

The proof is a simple adaptation from the proof of Theorem 7.

We now suggest a parameter setting for $(\mathcal{A}, \mathcal{B})$-Prod that guarantees constant regret against the benchmark $\mathcal{B}$ and $\mathcal{O}(\sqrt{T \log T})$ regret against the learning algorithm $\mathcal{A}$ in the worst–case.

Corollary 1. *Let* $C \ge 1$ *be an upper bound on the total expected benchmark loss* $\widehat{L}_T(\mathcal{B})$*. Then setting* $\eta = \frac{1}{2}\sqrt{\frac{\log C}{C}} < \frac{1}{2}$ *and* $\mathbf{w}_{\mathcal{B},1} = 1 - \mathbf{w}_{\mathcal{A},1} = 1 - \eta$ *simultaneously*

guarantees

$$\mathcal{R}_T\big((\mathcal{A},\mathcal{B})\text{-PROD},\mathbf{x}\big) \leq \mathcal{R}_T(\mathcal{A},\mathbf{x}) + 2\sqrt{C\log C}$$

for any $\mathbf{x}\in\mathcal{S}$ *and*

$$\mathcal{R}_T\big((\mathcal{A},\mathcal{B})\text{-PROD},\mathcal{B}\big) \leq 2\log 2$$

against any assignment of the loss sequence.

Proof. (Corollary 1) The second part follows from the fact that $\frac{\log(1-\eta)}{\eta}$ is a decreasing function on $\eta\in\big(0,\frac{1}{2}\big)$. For the first part, we study two cases. In the first case, we assume that $\widehat{L}_T(\mathcal{B}) \leq \widehat{L}_T(\mathcal{A})$ holds, which proves the statement for this case. For the second case, we assume $\widehat{L}_T(\mathcal{A}) \leq \widehat{L}_T(\mathcal{B})$ and notice that

$$\begin{aligned}
\sum_{t=1}^{T}\big(f_t(\mathbf{b}_t) - f_t(\mathbf{a}_t)\big)^2 &\leq \sum_{t=1}^{T}\left(f_t(\mathbf{b}_t)^2 + f_t(\mathbf{a}_t)^2\right)\\
&\leq \sum_{t=1}^{T}\left(f_t(\mathbf{b}_t) + f_t(\mathbf{a}_t)\right)\\
&\leq L_T(\mathcal{B}) + L_T(\mathcal{A})\\
&\leq 2L_T(\mathcal{B})\\
&\leq 2C.
\end{aligned}$$

Plugging this result into the first bound of Theorem 13 and substituting the choice of η gives the result. □

Notice that for any $\mathbf{x}\in\mathcal{S}$, the previous bounds can be written as

$$\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-PROD},\mathbf{x}) \leq \min\left\{\mathcal{R}_T(\mathcal{A},\mathbf{x}) + 2\sqrt{C\log C}, \mathcal{R}_T(\mathcal{B},\mathbf{x}) + 2\log 2\right\},$$

which states that $(\mathcal{A},\mathcal{B})$-PROD achieves the minimum between the regret of the benchmark $\mathcal{B}$ and learning algorithm $\mathcal{A}$ plus an additional regret of $\mathcal{O}(\sqrt{C\log C})$. If we consider that in most online optimization settings, the worst–case regret for a learning algorithm is $\mathcal{O}(\sqrt{T})$, (see, e.g., the expert setting studied in Sect. 5.1), the previous bound shows that at the cost of an additional factor of $\mathcal{O}(\sqrt{T\log T})$ in the worst–case, $(\mathcal{A},\mathcal{B})$-PROD performs as well as the benchmark, which is very useful whenever $\mathcal{R}_T(\mathcal{B},x)$ is small. This suggests that if we set $\mathcal{A}$ to a learning

Initialize: Experts $\{1, \dots, K\}$, Decision set $\mathcal{S}$, Learning rate $\eta_1 = \frac{1}{2}, w_{\mathcal{B},1} = (0,1), w_{\mathcal{A},1} = 1 - w_{\mathcal{B},1}$, Algorithms $\mathcal{A}$ and $\mathcal{B}$, Rounds T, Function class $\mathcal{F}$.
Initialize: $\mathbf{a}_1 = \mathcal{A}(\emptyset, U_1)$ and $\mathbf{b}_1 = \mathcal{B}(\emptyset, V_1)$
For all $t = 1, 2, \dots, T$**, repeat**

1. Learner updates $s_t = \frac{\eta_t w_{\mathcal{A},t}}{\eta_t w_{\mathcal{A},t} + \frac{w_{\mathcal{B},1}}{2}}$.
2. Simultaneously
   - Environment chooses $f_t \in \mathcal{F}$.
   - Learner predicts $\mathbf{x}_t = \begin{cases} \mathbf{a}_t & \text{with probability } s_t, \\ \mathbf{b}_t & \text{with probability } 1 - s_t. \end{cases}$
3. Environment reveals f_t.
4. Learner suffers loss $f_t(\mathbf{x}_t)$.
5. Learner observes $\mathbf{a}_t = \mathcal{A}(\{f_s\}_{s=1}^t, U_t)$ and $\mathbf{b}_t = \mathcal{B}(\{f_s\}_{s=1}^t, V_t)$.
6. Learner updates $\delta_t = f_t(\mathbf{a}_t) - f_t(\mathbf{b}_t)$.
7. Learner updates $\eta_{t+1} = \left(1 + \sum_{s=1}^t \delta_t^2\right)^{-\frac{1}{2}}$
8. Learner updates $w_{t+1,\mathcal{A}} = w_{t,\mathcal{A}} \left(1 - \eta_t \delta_t\right)^{\frac{\eta_{t+1}}{\eta_t}}$.

end for

Figure 4.9: $(\mathcal{A}, \mathcal{B})$-Prod (Anytime)

algorithm with worst–case guarantees and set $\mathcal{B}$ to any benchmark, then $(\mathcal{A}, \mathcal{B})$-Prod successfully manages the downside risk exposure of any problem by finding a suitable mixture of $\mathcal{A}$ and $\mathcal{B}$.

Finally, we note that the parameter proposed in Corollary 1 can hardly be computed in practice, since an upper-bound on the loss of the benchmark $\widehat{L}_T(\mathcal{B})$ is rarely available. Fortunately, we can adapt an improved version of Prod with adaptive learning rates recently proposed by Gaillard et al. [2014] and obtain an anytime version of $(\mathcal{A}, \mathcal{B})$-Prod.

Algorithm 4.9 presents the adaptation of the adaptive-learning-rate Prod variant recently proposed by Gaillard et al. [2014] to our setting. Following their analysis, we can prove the following performance guarantee concerning the adaptive version of $(\mathcal{A}, \mathcal{B})$-Prod.

Theorem 14. *Let $\widehat{L}_T(\mathcal{B})$ be the total benchmark loss. Then anytime $(\mathcal{A}, \mathcal{B})$-Prod*

simultaneously guarantees

$$\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-PROD},\mathbf{x}) \leq \mathcal{R}_T(\mathcal{A},\mathbf{x}) + K_T\sqrt{\widehat{L}_T(\mathcal{B})+1} + 2K_T$$

for any $\mathbf{x} \in \mathcal{S}$ *and*

$$\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-PROD},\mathcal{B}) \leq 2\log 2 + 2K_T$$

against any assignment of the loss sequence, where $K_T = \mathcal{O}(\log\log T)$.

There are some notable differences between the guarantees given by the above theorem and Theorem 13. The most important difference is that the current statement guarantees an improved regret of $\mathcal{O}(\sqrt{T}\log\log T)$ instead of $\sqrt{T\log T}$ in the worst–case. However, this comes at the price of an $\mathcal{O}(\log\log T)$ regret against the benchmark strategy.

4.2 Discussion

Here we provide some intuition why D-PROD works with the linear update in PROD and not with the exponential update in HEDGE. As discussed by Even-Dar et al. [2008], any *difference* algorithm such as Hedge, Prod and FPL that base decisions solely on the *cumulative difference* between $f_t(a_t)$ and $f_t(b_t)$ suffer an additional regret of $\mathcal{O}(\sqrt{T})$ on both $\mathcal{A}$ and $\mathcal{B}$. A similar observation has been made by de Rooij et al. [2014], who discuss the possibility of combining a robust learning algorithm and *FTL* by HEDGE and conclude that this approach is insufficient for their goals (see also Section 5.1). Difference algorithms (DA) achieve bicriteria bounds

$$\mathcal{R}_T(DA,\mathbf{x}) \leq \mathcal{O}(\sqrt{T}),$$

for any $x \in \mathcal{S}$ and

$$\mathcal{R}_T(DA,D) \leq \Omega(\sqrt{T}),$$

to a fixed uniform allocation over experts D. In the worst case, the product of these regrets is $\Omega(T)$. Even-Dar et al. [2008] noted that, gradually increasing expert weights, in favor of the expert showing improved performance, results in breaking this performance bottleneck. [Even-Dar et al., 2008] introduce a simple trick to achieve this momentum update. The second-order regret upper bound in Prod, to any individual expert, satisfies

$$\mathcal{R}_T(\text{Prod}, k) \leq \underbrace{\eta \sum_{t=1}^{T} l_{k,t}^2}_{\text{Expert-specific term}} + \frac{\log K}{\eta},$$

for any $\mathbf{x} \in \mathcal{S}$. By computing losses as the difference to the special expert D, a fixed distribution over experts $\{1, \ldots, K\}$,

$$\mathcal{R}_T(\text{Prod}, D) \leq \eta \sum_{t=1}^{T} (l_{i,t} - l_{D,t})^2 + \frac{\log K}{\eta},$$

[Even-Dar et al., 2008] satisfy the following regret bound to the fixed allocation D,

$$\mathcal{R}_T(\text{Prod}, D) \leq \frac{\log K}{\eta}.$$

By choosing a fixed expert D that is not updated with the *base* experts $\{1, \ldots, K\}$, D acts as an effective benchmark, enabling *D-Prod* to achieve an excellent bicriteria regret bound. This result is not possible with the first–order bounds in Hedge that contain a fixed first term, $\frac{\eta T}{8}$.

The impact of the first term results from the difference in updates. The Prod linear update $w_{i,t+1} = w_{i,t}(1 - \eta \ell_{i,t})$, is close to the exponential Hedge update $w_{i,t+1} = w_{i,t} \exp(-\eta \ell_{i,t})$, for small η, up to second-order quantities [Cesa-Bianchi et al., 2007]. This "approximate" exponential update results in the necessary momentum for resolving the limitation of difference algorithms in Theorem 11. The updates are illustrated in the following table inside of our proposed $(\mathcal{A}, \mathcal{B})$ adaptation structure, comparing $(\mathcal{A}, \mathcal{B})$-Prod, with update $w_{\mathcal{A},t+1} = w_{\mathcal{A},t}(1 - \eta(\ell_{\mathcal{A},t} - \ell_{\mathcal{B},t}))$, to an alternative exponential (Hedge) update in $(\mathcal{A}, \mathcal{B})$-Hedge, with update $w_{\mathcal{A},t+1} = w_{\mathcal{A},t} \exp(-\eta(\ell_{\mathcal{A},t} - \ell_{\mathcal{B},t}))$. We set losses $f_{\mathcal{A},t} = 0, 1, 0, \ldots, 1$

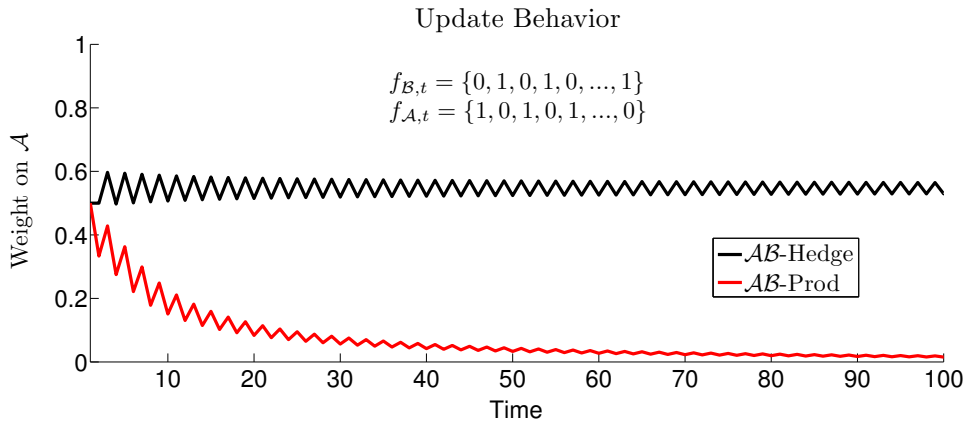

Figure 4.10: Impact of PROD update.

and $f_{\mathcal{B},t} = 1, 0, 1, \ldots, 0$, with a large starting weight in favor of the benchmark $\mathcal{B}$ and a learning rate set to $\eta = 0.5$. The following table illustrates the updates,

t	$f_{\mathcal{A},t}$	$(\mathcal{A},\mathcal{B})$-PROD	$(\mathcal{A},\mathcal{B})$-HEDGE
0	*initialize*	$\mathbf{w}_{\mathcal{A},0} = \frac{1}{K}$	$\mathbf{w}_{\mathcal{A},0} = \frac{1}{K}$
1	0	$\mathbf{w}_{\mathcal{A},1} = \frac{1}{K}\left(\frac{3}{2}\right) = \frac{3}{2K}$	$\mathbf{w}_{\mathcal{A},1} = \frac{1}{K}\exp\left(\frac{1}{2}\right)$
2	1	$\mathbf{w}_{\mathcal{A},2} = \frac{1}{K}\left(\frac{1}{2}\right)\left(\frac{3}{2}\right) = -\frac{3}{4K}$	$\mathbf{w}_{\mathcal{A},2} = \frac{1}{K}\exp\left(\frac{1}{2}\right)\exp\left(-\frac{1}{2}\right) = \frac{1}{K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	1	$\mathbf{w}_{\mathcal{A},100} = \frac{1}{K}\left(\frac{1}{2}\right)^{50}\left(\frac{3}{2}\right)^{50} \approx 0$	$\mathbf{w}_{\mathcal{A},100} = \exp\left(\frac{1}{2}\right)^{50}\exp\left(-\frac{1}{2}\right)^{50} = \frac{1}{K}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Due to the large starting weight in favor of benchmark $\mathcal{B}$, and the repeated series of $0, 1$ losses, $(\mathcal{A},\mathcal{B})$-PROD quickly prefers the benchmark $\mathcal{B}$, while $(\mathcal{A},\mathcal{B})$-HEDGE indecisively flips back and forth between the two experts. The result is further illustrated in Figure 4.10.

5 Applications

In this chapter, we defined an algorithm that provides a general structure that can be instantiated in a wide range of settings by simply plugging in the most appropriate choice of two algorithms. A straightforward application of the benchmark in $(\mathcal{A},\mathcal{B})$-PROD is a learning algorithm that exploits "easy" data sequences, while

providing worst-case guarantees on "hard" sequences. Given a learning algorithm $\mathcal{A}$, with worst-case performance guarantees, and a benchmark strategy $\mathcal{B}$, exploiting a specific structure within the loss sequence, $(\mathcal{A}, \mathcal{B})$-Prod smoothly adapts to "easy" and "hard" problems. $(\mathcal{A}, \mathcal{B})$-Prod achieves the best possible guarantees on both types of loss sequences, while providing the protection of worst-case guarantees on "hard" sequences. In the following subsections, we explore algorithms in disparate problem settings, where the benchmark is set to a problem specific learning algorithm that exploits the structure in easy data sequences as the $(\mathcal{A}, \mathcal{B})$-Prod benchmark.

5.1 Prediction with expert advice

de Rooij et al. [2014] note that prediction with expert advice algorithms are usually too conservative to exploit "easily learnable" loss sequences and might be significantly outperformed by *FTL* (see e.g., Figure 4.2), which exploits the structure of losses to achieve regret sublinear in K and is known to be optimal in the case of i.i.d. losses, where it achieves a regret of $\mathcal{O}(\log T)$. As a direct consequence of Corollary 1, we can use the general structure of $(\mathcal{A}, \mathcal{B})$-Prod to match the performance of *FTL* on "easy" data, and at the same time, obtain the same worst–case guarantees of standard algorithms for prediction with expert advice. In particular, if we set *FTL* as the benchmark $\mathcal{B}$ and AdaHedge (see de Rooij et al. [2014]) as the learning algorithm $\mathcal{A}$, we obtain the following.

Theorem 15. *Let $\mathcal{S} = \Delta_K$ and $\mathcal{F} = [0,1]^K$. Running $(\mathcal{A}, \mathcal{B})$-Prod with $\mathcal{A} =$ AdaHedge and $\mathcal{B} =$ FTL, with the parameter setting suggested in Corollary 1 simultaneously guarantees,*

$$\begin{aligned}\mathcal{R}_T((\mathcal{A}, \mathcal{B})\text{-Prod}, \mathbf{x}) &\leq \mathcal{R}_T(\text{AdaHedge}, \mathbf{x}) + 2\sqrt{C \log C} \\ &\leq \sqrt{\frac{L_T^*(T - L_T^*)}{T} \log K} + 2\sqrt{C \log C},\end{aligned}$$

for any $x \in \mathcal{S}$, where $L_T^* = \min_{x \in \Delta_N} L_T(x)$, and,

$$\mathcal{R}_T\big((\mathcal{A}, \mathcal{B})\text{-Prod}, \text{FTL}\big) \leq 2 \log 2,$$

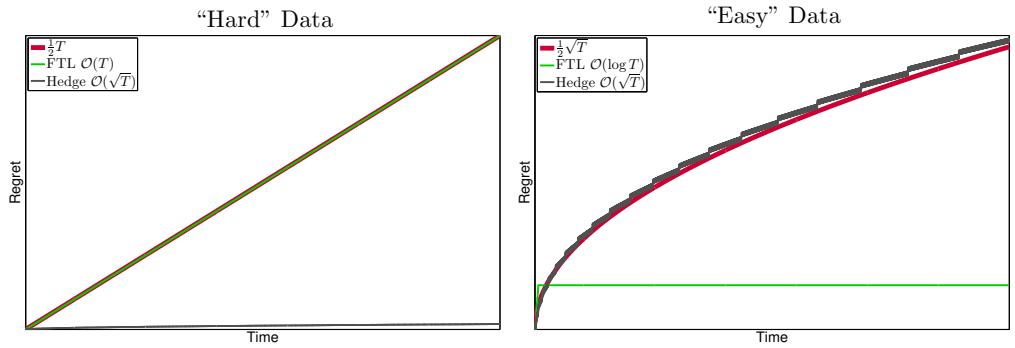

Figure 4.11: Performance comparison of *FTL* and *HEDGE* on easy versus hard data.

against any assignment of the loss sequence.

While we recover the worst–case guarantee of $\mathcal{O}(\sqrt{T \log K})$ plus an additional regret $\mathcal{O}(\sqrt{T \log T})$ on "hard" loss sequences, on "easy" problems we inherit the good performance of *FTL*. Note that a straightforward modification of *D*-PROD guarantees worst–case regret of $\mathcal{O}(\sqrt{C \log C \log K})$, which is asymptotically inferior to the guarantees given by Theorem 15. In the special case where the total loss of *FTL* and the regret of ADAHEDGE are equivalent to $\Theta(\sqrt{T})$, *D*-PROD guarantees a regret of $\mathcal{O}(T^{\frac{1}{4}})$, while the $(\mathcal{A}, \mathcal{B})$-PROD guarantee remains at $\mathcal{O}(\sqrt{T})$.

5.1.1 Comparison with FLIPFLOP

The FLIPFLOP algorithm proposed by de Rooij et al. [2014] addresses the problem of constructing algorithms that perform nearly as well as *FTL* on "easy" problems while retaining optimal guarantees on all possible loss sequences. More precisely, FLIPFLOP is a HEDGE algorithm where the learning rate η alternates between infinity (corresponding to *FTL*) and the value suggested by ADAHEDGE depending on the cumulative mixability gaps over the two regimes. The resulting algorithm is guaranteed to achieve the regret guarantees of

$$\mathcal{R}_T(\text{FLIPFLOP}, \mathbf{x}) \leq 5.64 \mathcal{R}_T(\text{FTL}, \mathbf{x}) + 3.73$$

and

$$\mathcal{R}_T(\text{FLIPFLOP}, \mathbf{x}) \leq 5.64 \sqrt{\frac{L_T^*(T - L_T^*)}{T} \log K} + \mathcal{O}(\log K)$$

against any fixed $\mathbf{x} \in \Delta_K$ at the same time. The latter bound is a so-called *second-order* regret bound with the property of being small whenever L_T^* is close to 0 or T. Regardless of the actual realization of losses, this result implies that the regret of FlipFlop is of optimal order.

Notice that while the guarantees in Theorem 15 are very similar in nature to those of de Rooij et al. [2014] concerning FlipFlop, the two results are slightly different. The worst–case bounds of $(\mathcal{A}, \mathcal{B})$-Prod are inferior by a factor of order $\sqrt{T \log T}$. In fact, the worst–case for our bound is realized when $C = \Omega(T)$, which is precisely the case when AdaHedge has excellent performance as it will be seen in Sect. 6. On the positive side, our guarantees are much stronger when *FTL* outperforms AdaHedge. To see this, observe that their regret bound can be rewritten as

$$L_T(\text{FlipFlop}) \leq L_T(\text{FTL}) + 4.64\big(L_T(\text{FTL}) - \inf_{\mathbf{x}} L_T(\mathbf{x})\big) + 3.73,$$

whereas our result replaces the last two terms by $2 \log 2$. While one can parametrize FlipFlop so as to decrease the gap between these bounds, the bound on $L_T(\text{FlipFlop})$ is always going to be linear in $\mathcal{R}_T(\text{FlipFlop}, x)$. The other advantage of our result is that we can directly bound the *total loss* of our algorithm in terms of the *total loss of* AdaHedge (see Theorem 13). This is to be contrasted with the result of de Rooij et al. [2014], who upper bound their *regret* in terms of the *regret bound* of AdaHedge, which may not be as tight and may be much worse in practice than the actual performance of AdaHedge. All these advantages of our approach stem from the fact that we smoothly mix the predictions of AdaHedge and *FTL*, while FlipFlop explicitly follows one policy or the other for extended periods of time, potentially accumulating unnecessary losses when switching too late or too early. Finally, we note that as FlipFlop is a sophisticated algorithm specifically designed for balancing the performance of AdaHedge and *FTL* in the expert setting, so we cannot reasonably expect to outperform it in every respect by using our general-purpose algorithm. Notice however that the analysis of FlipFlop is difficult to generalize to other learning

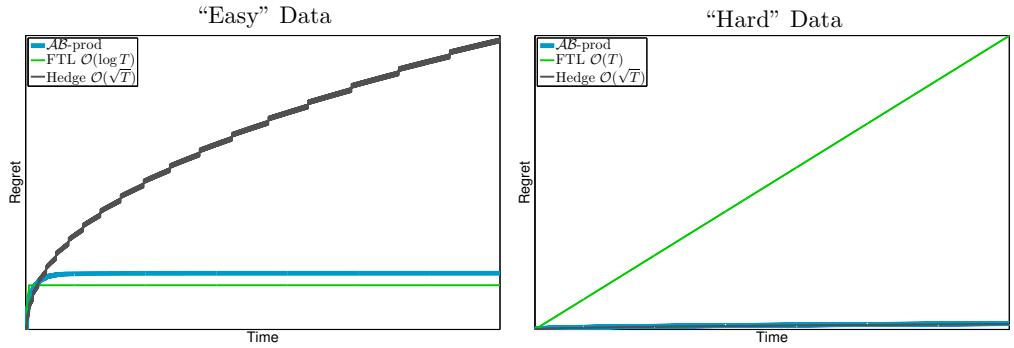

Figure 4.12: Performance comparison of $(\mathcal{A}, \mathcal{B})$-PROD, *FTL* and *HEDGE* on easy versus hard data.

settings such as the ones we discuss in the sections below.

5.2 Tracking the best expert

We now turn to the problem of tracking the best expert, where the goal of the learner is to control the regret against the best fixed strategy that is allowed to change its prediction at most S times during the entire decision process (see, e.g., Herbster and Warmuth [1998], György et al. [2012]). The regret of an algorithm $\mathcal{A}$ producing predictions $a_1, \ldots, a_T$ against an arbitrary sequence of decisions $y_{1:T} \in \mathcal{S}^T$ is defined as

$$\mathcal{R}_T(\mathcal{A}, y_{1:T}) = \sum_{t=1}^{T} \big(f_t(a_t) - f_t(y_t)\big).$$

Regret bounds in this setting typically depend on the complexity of the sequence $y_{1:T}$ as measured by the number decision switches

$$C(y_{1:T}) = \{t \in \{2, \ldots, T\} : y_t \neq y_{t-1}\}.$$

For example, a properly tuned version of the FIXED-SHARE (FS) algorithm of Herbster and Warmuth [1998] guarantees that

$$\mathcal{R}_T(FS, y_{1:T}) = \mathcal{O}\big(C(y_{1:T})\sqrt{T \log K}\big).$$

This upper bound can be tightened to $\mathcal{O}(\sqrt{ST \log K})$ when the learner knows an upper bound S on the complexity of $y_{1:T}$. While this bound is unimprovable in

general, one might wonder if it is possible to achieve better performance when the loss sequence is "easy". This precise question was posed very recently as a COLT open problem by Warmuth and Koolen [2014]. The generality of our approach allows us to solve their open problem by using $(\mathcal{A},\mathcal{B})$-PROD as a master algorithm to combine an opportunistic strategy with a principled learning algorithm. The following theorem states the performance of the $(\mathcal{A},\mathcal{B})$-PROD-based algorithm.

Theorem 16. *Let $\mathcal{S} = \Delta_K$, $\mathcal{F} = [0,1]^K$ and $y_{1:T}$ be any sequence in $\mathcal{S}$ with known complexity $S = C(y_{1:T})$. Running $(\mathcal{A},\mathcal{B})$-PROD with an appropriately tuned instance of $\mathcal{A} =$ FS (see Herbster and Warmuth [1998]), with the parameter setting suggested in Corollary 1 simultaneously guarantees*

$$\begin{aligned}\mathcal{R}_T\big((\mathcal{A},\mathcal{B})\text{-PROD}, y_{1:T}\big) &\le \mathcal{R}_T(FS, y_{1:T}) + 2\sqrt{C\log C}\\ &= \mathcal{O}(\sqrt{ST\log K}) + 2\sqrt{C\log C}\end{aligned}$$

and

$$\mathcal{R}_T\big((\mathcal{A},\mathcal{B})\text{-PROD}, \mathcal{B}\big) \le 2\log 2,$$

against any assignment of the loss sequence.

The remaining problem is then to find a benchmark that works well on "easy" problems, notably when the losses are i.i.d. in S (unknown) segments of the rounds $1,\dots,T$. Out of the strategies suggested by Warmuth and Koolen [2014], we analyze a windowed variant of *FTL* (referred to as FTL(w)) that bases its decision at time t on losses observed in the time window $[t-w-1, t-1]$ and picks expert $b_t = \arg\min_{\mathbf{x}\in\Delta_K} \mathbf{x}^\top \sum_{m=t-w-1}^{t-1} \ell_m$. The next proposition (proved in the appendix) gives a performance guarantee for FTL(w) with an optimal parameter setting.

Proposition 1. *Assume that there exists a partition of $[1,T]$ into S intervals $I_1,\dots,I_S$, such that the i-th component of the loss vectors within each interval I_s are drawn independently from a fixed probability distribution $\mathcal{D}_{s,i}$ dependent on the index s of the interval and the identity of expert i. Furthermore, assume that at any time t, there exists a unique expert i_t^* and gap parameter $\delta > 0$ such that $\mathbb{E}\left[\ell_{t,i_t^*}\right] \le \mathbb{E}\left[\ell_{t,i}\right] - \delta$ holds for all $i \neq i_t^*$. Then, the regret* FTL(w) *with parameter*

$w > 0$ is bounded as

$$\mathbb{E}\left[\mathcal{R}_T(\mathrm{FTL}(w), y_{1:T})\right] \leq wS + KT \exp\left(-\frac{w\delta^2}{4}\right),$$

where the expectation is taken with respect to the distribution of the losses. Setting

$$w = \left\lceil \frac{4 \log\left(\frac{KT}{S}\right)}{\delta^2} \right\rceil,$$

the bound becomes

$$\mathbb{E}\left[\mathcal{R}_T(\mathrm{FTL}(w), y_{1:T})\right] \leq \frac{4S \log\left(\frac{KT}{S}\right)}{\delta^2} + 2S.$$

Proof. The proof is based on upper bounding the probabilities $q_t = \mathbb{P}\left[b_t \neq i_t*\right]$ for all t. First, observe that the contribution of a round when $b_t = i_t^*$ to the expected regret is zero, thus the expected regret is upper bounded by $\sum_{t=1}^{T} q_t$. We say that t is in the w-interior of the partition if $t \in I_s$ and $t > \min\{I_s\} + w$ hold for some s, so that b_t is computed solely based on samples from $\mathcal{D}_s$. Let $\hat{\ell}_t = \sum_{m=t-w-1}^{t-1} \ell_m$ and $\bar{\ell}_t = \mathbb{E}\left[\ell_t\right]$. By Hoeffding's inequality, we have that

$$\begin{aligned} q_t = \mathbb{P}\left[b_t \neq i_t^*\right] &\leq \mathbb{P}\left[\exists i : \hat{\ell}_{i_t^*,t} > \hat{\ell}_{i,t}\right] \\ &\leq \sum_{i=1}^{K} \mathbb{P}\left[\left(\bar{\ell}_{i,t} - \bar{\ell}_{i_t^*,t}\right) - \left(\hat{\ell}_{i,t} - \hat{\ell}_{i_t^*,t}\right) > \delta\right] \\ &\leq K \exp\left(-\frac{w\delta^2}{4}\right) \end{aligned}$$

holds for any t in the w-interior of the partition. The proof is concluded by observing that there are at most wS rounds outside the w-interval of the partition and using the trivial upper bound on q_t on such rounds. □

5.3 Online convex optimization

Here we consider the problem of online convex optimization (OCO), where $\mathcal{S}$ is a convex and closed subset of $\mathbb{R}^K$ and $\mathcal{F}$ is the family of convex functions on $\mathcal{S}$. In this setting, if we assume that the loss functions are smooth (see Zinkevich

[2003]), an appropriately tuned version of the online gradient descent (*OGD*) is known to achieve a regret of $\mathcal{O}(\sqrt{T})$. As shown by Hazan et al. [2007a], if we additionally assume that the environment plays *strongly convex* loss functions and tune the parameters of the algorithm accordingly, the same algorithm can be used to guarantee an improved regret of $\mathcal{O}(\log T)$. Furthermore, they also show that *FTL* enjoys essentially the same guarantees. Hazan et al. [2007b] studied whether the two guarantees could be combined. They present the adaptive online gradient descent (*AOGD*) algorithm that guarantees $\mathcal{O}(\log T)$ regret when the aggregated loss functions $F_t = \sum_{s=1}^{t} f_s$ are strongly convex for all t, while retaining the $\mathcal{O}(\sqrt{T})$ bounds if this is not the case. The next theorem shows that we can replace their complicated analysis by our general argument and show essentially the same guarantees.

Theorem 17. *Let $\mathcal{S}$ be a convex closed subset of $\mathbb{R}^K$ and $\mathcal{F}$ be the family of smooth convex functions on $\mathcal{S}$. Running $(\mathcal{A},\mathcal{B})$-Prod with an appropriately tuned instance of $\mathcal{A} = \mathrm{OGD}$ (see Zinkevich [2003]) and $\mathcal{B} = \mathrm{FTL}$, with the parameter setting suggested in Corollary 1 simultaneously guarantees*

$$\begin{aligned}\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-Prod},\mathbf{x}) &\leq \mathcal{R}_T(OGD,\mathbf{x}) + 2\sqrt{C\log C}\\ &= \mathcal{O}(\sqrt{T}) + 2\sqrt{C\log C}\end{aligned}$$

for any $\mathbf{x}\in\mathcal{S}$ and

$$\mathcal{R}_T\big((\mathcal{A},\mathcal{B})\text{-Prod},\mathrm{FTL}\big) \leq 2\log 2.$$

against any assignment of the loss sequence. In particular, this implies that

$$\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-Prod},\mathbf{x}) = \mathcal{O}(\log T)$$

if the loss functions are strongly convex.

Similar to the previous settings, at the cost of an additional regret of $\mathcal{O}(\sqrt{T\log T})$ in the worst–case, $(\mathcal{A},\mathcal{B})$-Prod successfully adapts to the "easy" loss sequences, which in this case corresponds to strongly convex functions, on which it achieves a $\mathcal{O}(\log T)$ regret. Notice that the same guarantees may be ob-

tained with any other pair of online convex optimization algorithms with similar properties (e.g., replacing *FTL* by the Online Newton Step method or OGD with a $\frac{1}{Ht}$ step size [Hazan et al., 2007a].).

5.4 Learning with two-points-bandit feedback

We consider the multi-armed bandit problem with two-point feedback. This is a special case of the partial-information game recently studied by Seldin et al. [2014]. A similar model has also been studied as a simplified version of online convex optimization with partial feedback [Agarwal et al., 2010]. We assume that in each round t, the learner picks one arm I_t in the decision set $\mathcal{S} = \{1, 2, \ldots, K\}$ and *also has the possibility to choose and observe the loss of another arm* J_t. The learner suffers the loss $f_t(I_t)$. Unlike the settings considered in the previous sections, the learner only gets to observe the loss function for arms I_t and J_t. While this setting does not entirely conform to our assumptions concerning $\mathcal{A}$ and $\mathcal{B}$, observe that a hedging strategy $\mathcal{C}$ defined over $\mathcal{A}$ and $\mathcal{B}$ only requires access to *the losses suffered by the two algorithms and not the entire loss functions*. Formally, we give $\mathcal{A}$ and $\mathcal{B}$ access to the decision set $\mathcal{S}$, and $\mathcal{C}$ to $\mathcal{S}^2$. The hedging strategy $\mathcal{C}$ selects the pair (I_t, J_t) based on the arms suggested by $\mathcal{A}$ and $\mathcal{B}$ as:

$$(I_t, J_t) = \begin{cases} (a_t, b_t) & \text{with probability } s_t, \\ (b_t, a_t) & \text{with probability } 1 - s_t. \end{cases}$$

The probability s_t is a well-defined deterministic function of $\mathcal{H}^*_{t-1}$, thus the regret bound of $(\mathcal{A}, \mathcal{B})$-Prod can be directly applied. In this case, "easy" problems correspond to i.i.d. loss sequences (with a fixed gap between the expected losses), for which the UCB algorithm of Auer et al. [2002] is guaranteed to have a $\mathcal{O}(\log T)$ regret, while on "hard" problems, we can rely on the Exp3 algorithm of Auer et al. [2002] which suffers a regret of $\mathcal{O}(\sqrt{TK})$ in the worst–case. The next theorem gives the performance guarantee of $(\mathcal{A}, \mathcal{B})$-Prod when combining UCB and Exp3.

Theorem 18. *Consider the multi-armed bandit problem with K arms and two-point feedback. Running $(\mathcal{A}, \mathcal{B})$-Prod with an appropriately tuned instance of*

$\mathcal{A} = \text{EXP3}$ *(see Auer et al. [1995]) and* $\mathcal{B} = \text{UCB}$ *(see Auer et al. [2002]), with the parameter setting suggested in Corollary 1 simultaneously guarantees*

$$\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-PROD},\mathbf{x}) \leq \mathcal{R}_T(\text{EXP3},\mathbf{x}) + 2\sqrt{C\log C} = \mathcal{O}(\sqrt{TK\log K}) + 2\sqrt{C\log C}$$

for any arm $\mathbf{x} \in \{1,2,\ldots,K\}$ *and*

$$\mathcal{R}_T\big((\mathcal{A},\mathcal{B})\text{-PROD},\text{UCB}\big) \leq 2\log 2.$$

against any assignment of the loss sequence. In particular, if the losses are generated in an i.i.d. fashion and there exists a unique best arm $\mathbf{x}^* \in \mathcal{S}$*, then*

$$\mathbb{E}\big[\mathcal{R}_T((\mathcal{A},\mathcal{B})\text{-PROD},\mathbf{x})\big] = \mathcal{O}(\log T),$$

where the expectation is taken with respect to the distribution of the losses.

This result shows that even in the multi-armed bandit setting, we can achieve nearly the best performance in both "hard" and "easy" problems given that we are allowed to pull two arms at the time. This result is to be contrasted with those of Bubeck and Slivkins [2012], who consider the standard one-point feedback setting where only a single evaluation in each round is allowed. They propose an algorithm called *SAO* that uses a very sophisticated decision rule to switch between an aggressive UCB-like strategy to the more safe EXP3 in case of "hard" loss sequences. The heavily technical analysis of Bubeck and Slivkins [2012] shows that *SAO* achieves $\mathcal{O}(\log^2 T)$ regret in stochastic environments and $\mathcal{O}(\sqrt{T}\log^{\frac{3}{2}} T)$ regret in the adversarial setting. While our result holds under stronger assumptions, Theorem 18 shows that $(\mathcal{A},\mathcal{B})$-PROD is not restricted to work only in full-information settings. Once again, we note that such a result cannot be obtained by simply combining the predictions of UCB and EXP3 by a generic learning algorithm as HEDGE. An algorithm designed specifically for the one–armed bandit setting is EXP3++ [Seldin and Slivkins, 2014], which is a variant of the EXP3 algorithm that simultaneously guarantees $\mathcal{O}(\log^2 T)$ regret in the stochastic environment, while retaining the regret bound of $\mathcal{O}(\sqrt{TK\log K})$ in the adversarial.

6 Empirical Results

We study the performance of $(\mathcal{A}, \mathcal{B})$-Prod in the experts setting to verify the theoretical results of Theorem 15, show the importance of the $(\mathcal{A}, \mathcal{B})$-Prod update rule and compare its performance to FlipFlop. We report performance results for *FTL*, AdaHedge, FlipFlop, AdaNormalHedge [Luo and Schapire, 2015], an anytime version of *D*-Prod, $(\mathcal{A}, \mathcal{B})$-Hedge, a variant of $(\mathcal{A}, \mathcal{B})$-Prod where an exponential weighting scheme is used, and both finite and anytime versions of $(\mathcal{A}, \mathcal{B})$-Prod. While the original *D*-Prod is designed for the finite time setting, we extend it to the adaptive–learning–rate Prod variant recently proposed by Gaillard et al. [2014] and also replace the fixed "special" expert D in its original design with *FTL*. Both $(\mathcal{A}, \mathcal{B})$-Prod, and $(\mathcal{A}, \mathcal{B})$-Hedge are set with $\mathcal{B} = \text{FTL}$ and $\mathcal{A} = \text{AdaHedge}$. Algorithms are evaluated on the datasets proposed by de Rooij et al. [2014], where deterministic data involving two experts is designed to illustrate four particular cases. In each case, data consists of an initial hand-crafted loss vector, followed by a sequence of 1999 loss vectors of either $(0, 1)$ or $(1, 0)$. The data are generated by sequentially appending the loss vector to bring the cumulative loss difference $L_{1,t} - L_{2,t}$ closer to a target function $f_\psi(t)$, where $\psi \in \{1, 2, 3, 4\}$ indexes a particular experiment. Each $f_\psi : [0, \infty) \to [0, \infty)$ is a nondecreasing function with $f_\psi(0) = 0$. Intuitively, it expresses how much better expert 2 is compared to expert 1, as a function of time. The functions f_ψ change slowly enough that it has the property $|L_{1,t} - L_{2,t} - f_\psi(t)| \leq 1$ for all t. For more details on each of these settings, please refer to de Rooij et al. [2014]. Results for the following configurations are reported in Figure 4.13.

6.1 Settings

Setting 1. This setting illustrates the worst case performance for *FTL*. It is defined by $l_1 = (\frac{1}{2}, 0)$ and $f_1(t) = (0)$. This results in the following loss matrix,

$$\begin{pmatrix} \frac{1}{2} & 0 & 1 & 0 & 1 & \dots \\ 0 & 1 & 0 & 1 & 0 & \dots \end{pmatrix}^\top$$

Setting 2. This setting illustrates the best case performance for *FTL*. It is defined by $l_1 = (\frac{1}{2}, 0)$ and $f_2(t) = (0)$. This results in the following loss matrix,

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 0 & 1 & \dots \end{pmatrix}^\top$$

Setting 3. This setting illustrates when weights do not concentrate in AdaHedge. It is defined by $l_1 = (\frac{1}{2}, 0)$ and $f_3(t) = t^{0.4}$. The first few loss vectors are equivalent to the setting in Experiment 2, but loss vectors are repeated on occasion. This causes a small performance gap between the experts.

Setting 4. This setting illustrates when weights concentrate in AdaHedge. This experiment is defined by $l_1 = (1, 0)$ and $f_4(t) = t^{0.6}$. This experiment is similar to the setting in Experiment 3, but with a larger performance gap between experts.

First, notice that $(\mathcal{A}, \mathcal{B})$-Prod always performs comparably with the best algorithm between $\mathcal{A}$ and $\mathcal{B}$. In setting 1, although *FTL* suffers linear regret, $(\mathcal{A}, \mathcal{B})$-Prod rapidly adjusts the weights towards AdaHedge and finally achieves the same order of performance. In settings 2 and 3, the situation is reversed since *FTL* has a constant regret, while AdaHedge has regret of order $\mathcal{O}(\sqrt{T})$. In this case, after a short initial phase where $(\mathcal{A}, \mathcal{B})$-Prod has an increasing regret, it stabilizes on the same performance as *FTL*. In setting 4, both AdaHedge and *FTL* have a constant regret and $(\mathcal{A}, \mathcal{B})$-Prod attains the same performance. These results match the behavior predicted in the bound of Theorem 15, which guarantees that the regret of $(\mathcal{A}, \mathcal{B})$-Prod is roughly the minimum of *FTL* and AdaHedge.

As discussed in Section 3, the Prod update rule used in $(\mathcal{A}, \mathcal{B})$-Prod plays

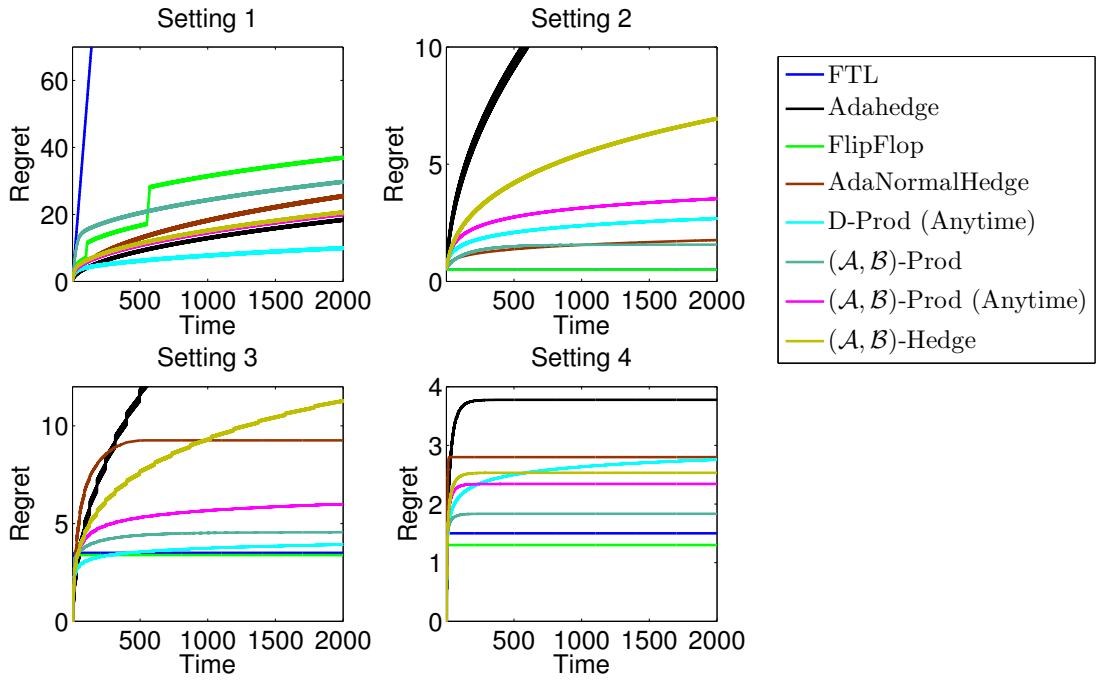

Figure 4.13: Hand tuned loss sequences from de Rooij et al. [2014]

a crucial role to obtain a constant regret against the benchmark, while other rules, such as the exponential update used in $(\mathcal{A},\mathcal{B})$-Hedge, may fail in finding a suitable mix between $\mathcal{A}$ and $\mathcal{B}$. As illustrated in settings 2 and 3, $(\mathcal{A},\mathcal{B})$-Hedge suffers a regret similar to AdaHedge and it fails to take advantage of the good performance of *FTL*, which has a constant regret. In setting 1, $(\mathcal{A},\mathcal{B})$-Hedge performs as well as $(\mathcal{A},\mathcal{B})$-Prod because *FTL* is constantly worse than AdaHedge and its corresponding weight is decreased very quickly, while in setting 4 both *FTL* and AdaHedge achieves a constant regret and so does $(\mathcal{A},\mathcal{B})$-Hedge. Finally, we compare $(\mathcal{A},\mathcal{B})$-Prod and FlipFlop. As discussed in Section 3, the two algorithms share similar theoretical guarantees with potential advantages of one on the other depending on the specific setting. In particular, FlipFlop performs slightly better in settings 2, 3, and 4, whereas $(\mathcal{A},\mathcal{B})$-Prod obtains smaller regret in setting 1, where the constants in the FlipFlop bound show their teeth. While it is not possible to clearly rank the two algorithms, $(\mathcal{A},\mathcal{B})$-Prod clearly avoids the pathological behavior exhibited by FlipFlop in setting 1. Finally, we note that the anytime version of *D*-Prod is slightly better than

$(\mathcal{A}, \mathcal{B})$-Prod, but no consistent difference is observed.

7 Conclusions

This chapter introduced $(\mathcal{A}, \mathcal{B})$-Prod. $(\mathcal{A}, \mathcal{B})$-Prod uses a flexible protection mechanism to enhance decision-theoretic risk tools in online learning. Further, $(\mathcal{A}, \mathcal{B})$-Prod guarantees order-optimal regret bounds, *while also guaranteeing a cumulative loss within a constant factor of some pre-defined benchmark*. We stress that this property is much stronger than simply guaranteeing $\mathcal{O}(1)$ regret with respect to some fixed distribution D, as done by Even-Dar et al. [2008]. $(\mathcal{A}, \mathcal{B})$-Prod allows comparisons to *any fixed, changing or adaptive benchmark*. This property is very important in practical risk-sensitive settings. In particular, $(\mathcal{A}, \mathcal{B})$-Prod can replace any existing solution at a (guaranteed) negligible cost in output performance, with additional strong guarantees in the worst-case. We showed that whenever $\mathcal{A}$ is a learning algorithm with worst-case performance guarantees and $\mathcal{B}$ is an opportunistic strategy exploiting a specific structure within the loss sequence, we obtain an algorithm which smoothly adapts to "easy" and "hard" problems.

Chapter 5

Conclusion

This thesis introduces novel algorithms for considering decision-theoretic risks in machine learning. An algorithm is presented for the accurate estimation of any statistic for any process, when only a short dependent sequence of observations are available, risk-averse algorithms are introduced to the multi-arm bandit setting, and finally, a flexible algorithm is introduced to provide a principled and flexible structure to "hedge" risk in the adversarial full-information setting.

First, the problem of accurate statistical estimation on a single short dependent time series sequence is considered. A novel information-theoretic iterative Bootstrap algorithm $\mathcal{R}$-Boot is introduced based on the *replacement Bootstrap principle*. This newly introduced principle generates bootstrap sequences by simultaneously replacing symbols using an estimated replacement distribution. $\mathcal{R}$-Boot is successfully demonstrated on both synthetic and real datasets. Preliminary theoretical and empirical results suggest that the replacement Bootstrap can significantly improve the estimation of complicated statistics in the general class of stationary–ergodic processes.

Next, a novel multi-armed bandit setting is introduced, where the objective is to perform as well as the arm with the best risk–return trade-off. The decision-theoretic risk associated to the variance over multiple runs and risk of variability associated to a single run of an algorithm are studied. Two algorithms are introduced, *MV-LCB* and *ExpExp*, with theoretical results to solve the Mean–Variance bandit problem. While *MV-LCB* shows a small regret of order $\mathcal{O}\left(\frac{\log T}{T}\right)$ on "easy" problems (i.e., where the Mean–Variance gaps Δ are big w.r.t. T), we showed that it has a constant worst–case regret. On the other hand, we proved that *ExpExp* has a vanishing worst–case regret at the cost of worse performance on "easy" problems. This is the first work to study risk–aversion in the stochastic multi–armed

bandit setting.

Finally, the problem of introducing a flexible and intuitive risk-averse structure is considered in the adversarial full-information setting. We introduce $(\mathcal{A}, \mathcal{B})$-PROD as a flexible protection mechanism to enhance risk-averse tools in online learning. Order-optimal regret bounds are provided, *while also guaranteeing a cumulative loss within a constant factor of some pre-defined benchmark.* This allows comparisons to *any fixed, changing or adaptive benchmark, that can optionally learn.* This allows $(\mathcal{A}, \mathcal{B})$-PROD to replace any existing online decision-making algorithm at a (guaranteed) negligible cost in performance, with additional strong guarantees in the worst-case. Results are provided in several problem settings. This thesis successfully introduces several ways to consider decision-theoretic risk in machine learning. Future works should consider extending these results in both application and theory. Though three specific settings are considered, many other settings should be extended to consider risk. Many areas consider risk as centrally important to decision-making.

Bibliography

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In A.T. Kalai and M. Mohri, editors, *Proceedings of the 23rd Annual Conference on Learning Theory*, pages 28–40, 2010. (Cited on page 121.)

Kareem Amin, Michael Kearns, Peter Key, and Anton Schwaighofer. Budget optimization for sponsored search: Censored learning in mdps. *arXiv preprint arXiv:1210.4847*, 2012. (Cited on page 50.)

András Antos, Varun Grover, and Csaba Szepesvári. Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411:2712–2728, June 2010. (Cited on page 65.)

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012. (Cited on page 93.)

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, (June 1996):1–24, 1999. (Cited on pages 84 and 85.)

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410:1876–1902, 2009. (Cited on pages 51, 52, 55, 56, 58, 66, 67, 82 and 86.)

Jean-Yves Audibert, Sébastien Bubeck, et al. Best arm identification in multi-armed bandits. *COLT 2010-Proceedings*, 2010. (Cited on pages 72 and 77.)

Peter Auer. Using upper confidence bounds for online learning. In *Proceedings of the 41th Annual Symposium on Foundations of Computer Science*, pages 270–293. IEEE Computer Society, 2000. (Cited on page 65.)

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995. (Cited on page 122.)

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. (Cited on pages 51, 52, 54, 55, 121 and 122.)

Orly Avner, Shie Mannor, and Ohad Shamir. Decoupling exploration and exploitation in multi-armed bandits. *arXiv preprint arXiv:1205.2874*, 2012. (Cited on page 50.)

Bilal M Ayyub. *Risk analysis in engineering and economics*. CRC Press, 2014. (Cited on page 1.)

Kevin E Bassler, Joseph L McCauley, and Gemunu H Gunaratne. Nonstationary increments, scaling distributions, and variable diffusion processes in financial markets. *Proceedings of the National Academy of Sciences*, 104(44):17287–17290, 2007. (Cited on page 45.)

Jeremy Berkowitz and Lutz Kilian. Recent developments in bootstrapping time series. *Econometric Reviews*, 19(1):1–48, 2000. (Cited on page 10.)

Joel Bessis. *Risk management in banking*. John Wiley & Sons, 2011. (Cited on page 1.)

Arup Bose and DN Politis. A review of the bootstrap for dependent samples. *Stochastic processes and statistical inference*, 1992. (Cited on page 10.)

David B. Brown. Large deviations bounds for estimating conditional value-at-risk. *Operations Research Letters*, 35:722–730, 2007. (Cited on page 85.)

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012. (Cited on page 51.)

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *COLT*, pages 42.1–42.23, 2012. (Cited on page 122.)

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006. (Cited on page 48.)

Peter Bühlmann. Bootstraps for time series. *Statistical Science*, 17(1):52–72, May 2002. (Cited on page 10.)

Peter Bühlmann et al. Sieve bootstrap for time series. *Bernoulli*, 3(2):123–148, 1997. (Cited on page 41.)

Oleg V Bychuk and Brian Haughey. *Hedging market exposures: Identifying and managing market risks*, volume 548. John Wiley & Sons, 2011. (Cited on page 3.)

Edward Carlstein. The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *The Annals of Statistics*, pages 1171–1179, 1986. (Cited on page 17.)

Edward Carlstein, Kim-Anh Do, Peter Hall, Tim Hesterberg, and Hans R. Kunsch. Matched-block bootstrap for dependent data. *Bernoulli*, 4(3):305–328, September 1998. (Cited on page 17.)

Alessandro Casati and Serge Tabachnik. The statistics of the maximum drawdown in financial time series. *Advances in Financial Risk Management: Corporates, Intermediaries and Portfolios*, page 347, 2013. (Cited on page 4.)

Nicolò Cesa-Bianchi and Gábor Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003. (Cited on page 51.)

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. (Cited on pages 88, 93, 94 and 96.)

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007. (Cited on pages 96 and 112.)

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011. (Cited on page 51.)

Chris Chatfield. *The analysis of time series: an introduction*. CRC press, 2013. (Cited on page 9.)

W Henry Chiu. Complete mean-variance preferences. 2007. (Cited on page 3.)

Philippe Cogneau and Valeri Zakamouline. Bootstrap methods for finance: Review and analysis. Technical report, HEC Management School, 2010. (Cited on page 15.)

Michel Crouhy, Dan Galai, and Robert Mark. *The Essentials of Risk Management, Second Edition*. McGraw-Hill Education, 2014. (Cited on page 1.)

Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Accepted to the Journal of Machine Learning Research*, 2014. (Cited on pages xii, 90, 111, 114, 115, 116, 123 and 125.)

Mark S Dorfman and David Cather. *Introduction to risk management and insurance*. Pearson Higher Ed, 2012. (Cited on page 1.)

Bradley Efron. Bootstrap methods: Another look at the Jackknife. *The Annals of Statistics*, 1979. (Cited on pages 10 and 15.)

Eyal Even-Dar, Michael Kearns, and Jennifer Wortman. Risk-sensitive online learning. In *Proceedings of the 17th international conference on Algorithmic Learning Theory (ALT'06)*, pages 199–213, 2006. (Cited on pages xii, 89, 90, 99, 100 and 101.)

Eyal Even-Dar, Michael Kearns, Yishay Mansour, and Jennifer Wortman. Regret to the best vs. regret to the average. *Machine Learning*, 72(1-2):21–37, 2008. (Cited on pages xii, 90, 103, 104, 106, 108, 111, 112 and 126.)

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997. (Cited on page 93.)

Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–9. IEEE, 2010. (Cited on page 50.)

Pierre Gaillard, Gilles Stoltz, and Tim van Erven. A second-order bound with excess losses. pages 176–196, 2014. (Cited on pages 110 and 123.)

Nicolas Galichet, Michèle Sebag, and Olivier Teytaud. Exploration vs exploitation vs safety: Risk-averse multi-armed bandits. *Journal of Machine Learning Research*, 29:245–260, 2013. (Cited on page 85.)

Itzhak Gilboa. *Theory of decision under uncertainty*, volume 45. Cambridge university press, 2009. (Cited on page 2.)

Rica Gonen and Elan Pavlov. An incentive-compatible multi-armed bandit mechanism. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 362–363. ACM, 2007. (Cited on page 50.)

Robert Gray. *Probability, Random Processes, and Ergodic Properties*. Springer Verlag, 1988. (Cited on page 13.)

Richard Grinold and Ronald Kahn. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Selecting Superior Returns and Controlling Risk*. McGraw-Hill Library of Investment and Finance. McGraw-Hill Education, 1999. (Cited on page 1.)

Peter Grunwald, Wouter M. Koolen, and Alexander Rakhlin, editors. *NIPS Workshop on "Learning faster from easy data"*, 2013. (Cited on page 90.)

András György, Tamás Linder, and Gábor Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012. (Cited on page 117.)

Peter Hall and Qiwei Yao. Data tilting for time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):425–442, 2003. (Cited on page 17.)

Peter Hall, Joel L Horowitz, and Bing-Yi Jing. On blocking rules for the bootstrap with dependent data. *Biometrika*, 82(3):561–574, 1995. (Cited on pages 17 and 41.)

James Hannan. Approximation to Bayes risk in repeated play. *Contributions to the theory of games*, 3:97–139, 1957. (Cited on page 93.)

Wolfgang Härdle, Joel Horowitz, and Jens-Peter Kreiss. Bootstrap methods for time series. *International Statistical Institute*, 71(2):435–459, August 2003. (Cited on page 10.)

Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009. (Cited on page 2.)

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69:169–192, 2007a. (Cited on pages 93, 120 and 121.)

Elad Hazan, Alexander Rakhlin, and Peter L Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pages 65–72, 2007b. (Cited on page 120.)

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998. (Cited on pages 117 and 118.)

GS Hongyi Li and Maddala. Bootstrapping time series models. *Econometric reviews*, 15(2):115–158, 1996. (Cited on page 10.)

Joel L Horowitz. The bootstrap. *Handbook of econometrics*, 5:3159–3228, 2001. (Cited on pages 16, 17 and 18.)

Joel L Horowitz. Bootstrap methods for Markov processes. *Econometrica*, 71(4): 1049–1082, 2003. (Cited on pages 11 and 19.)

X Hu, B Zhang, W Liu, S Paciga, W He, TA Lanz, R Kleiman, B Dougherty, SK Hall, AM McIntosh, et al. A survey of rare coding variants in candidate genes in schizophrenia by deep sequencing. *Molecular psychiatry*, 19(8):858–859, 2014. (Cited on page 10.)

Hull. *Risk Management and Financial Institutions,+ Web Site*, volume 733. John Wiley & Sons, 2012. (Cited on page 1.)

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. (Cited on page 93.)

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012. (Cited on page 51.)

Frank H Knight. Risk, uncertainty and profit. *New York: Hart, Schaffner and Marx*, 1921. (Cited on page 2.)

Frank H Knight. *Risk, uncertainty and profit*. Courier Corporation, 2012. (Cited on page 1.)

Danxia Kong and Maytal Saar-Tsechansky. Collaborative information acquisition for data-driven decisions. *Machine learning*, 95(1):71–86, 2014. (Cited on page 1.)

Jens-Peter Kreiss and Soumendra Nath Lahiri. Bootstrap methods for time series. *Handbook of Statistics: Time Series Analysis: Methods and Applications*, 30:1, 2012. (Cited on pages 10 and 16.)

Rafail Evseevich Krichevsky. A relation between the plausibility of information about a source and encoding redundancy. *Problems Information Transmission*, 4(3):48–57, 1968. (Cited on page 24.)

Hans R Kuensch. Statistical aspects of self-similar processes. In *Proceedings of the first World Congress of the Bernoulli Society*, volume 1, pages 67–74. VNU Science Press Utrecht, 1987. (Cited on page 17.)

RJ Kulperger and BLS Prakasa Rao. Bootstrapping a finite state Markov chain. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 178–191, 1989. (Cited on page 10.)

Soumendra Nath Lahiri. On the moving block bootstrap under long range dependence. *Statistics & Probability Letters*, 18(5):405–413, 1993. (Cited on page 18.)

Soumendra Nath Lahiri. Theoretical comparisons of block bootstrap methods. *Annals of Statistics*, pages 386–404, 1999. (Cited on pages 10, 17 and 18.)

Soumendra Nath Lahiri. *Resampling methods for dependent data*. Springer, 2003. (Cited on page 10.)

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985. (Cited on pages 51 and 54.)

James Lam. *Enterprise risk management: from incentives to controls*. John Wiley & Sons, 2014. (Cited on page 1.)

Olivier Ledoit and Michael Wolf. Honey, i shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004. (Cited on page 103.)

Haim Levy and Harry M Markowitz. Approximating expected utility by a function of mean and variance. *The American Economic Review*, pages 308–317, 1979. (Cited on page 3.)

Zhidong Li, Bang Zhang, Yang Wang, Fang Chen, Ronnie Taib, Vicky Whiffin, and Yi Wang. Water pipe condition assessment: a hierarchical beta process approach for sparse incident data. *Machine learning*, 95(1):11–26, 2014. (Cited on page 1.)

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994. (Cited on page 93.)

Regina Y Liu and Kesar Singh. Moving blocks jackknife and bootstrap capture weak dependence. *Exploring the limits of bootstrap*, 225:248, 1992. (Cited on page 17.)

H. Luo and R. E. Schapire. Achieving All with No Parameters: Adaptive NormalHedge. *ArXiv e-prints*, February 2015. (Cited on page 123.)

James G MacKinnon. Bootstrap methods in econometrics*. *Economic Record*, 82 (s1):S2–S18, 2006. (Cited on page 16.)

Odalric-Ambrym Maillard. Robust risk-averse stochastic multi-armed bandits. In *Algorithmic Learning Theory*, pages 218–233. Springer, 2013. (Cited on page 86.)

Benoit B. Mandelbrot and John W. van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM Review*, 10:422–437, 1968. (Cited on page 40.)

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. (Cited on pages 2, 50, 51, 57, 83 and 89.)

Harry Markowitz. Mean–variance approximations to expected utility. *European Journal of Operational Research*, 234(2):346–355, 2014. (Cited on page 3.)

Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The Annals of Probability*, 18(3):pp.1269–1283, 1990. ISSN 00911798. (Cited on page 84.)

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample-variance penalization. In *Proceedings of the Twenty-Second Annual Conference on Learning Theory*, pages 115–124, 2009. (Cited on page 67.)

Amy McGovern, David J Gagne Ii, John K Williams, Rodger A Brown, and Jeffrey B Basara. Enhancing understanding and improving prediction of severe weather through spatiotemporal relational learning. *Machine Learning*, 95(1): 27–50, 2014. (Cited on page 1.)

Aditya Krishna Menon, Xiaoqian Jiang, Jihoon Kim, Jaideep Vaidya, and Lucila Ohno-Machado. Detecting inappropriate access to electronic health records using collaborative filtering. *Machine learning*, 95(1):87–101, 2014. (Cited on page 1.)

Gusztáv Morvai, Sidney Yakowitz, and László Györfi. Nonparametric inference for ergodic, stationary time series. *The Annals of Statistics*, 24(1):370–379, 1996. (Cited on page 38.)

John von Neumann and Oskar Morgenstern. Theory of games and economic behavior. *Princeton University, Princeton*, 1947. (Cited on page 2.)

Daniel J Nordman et al. A note on the stationary bootstrap's variance. *The Annals of Statistics*, 37(1):359–370, 2009. (Cited on page 17.)

Timothy O'Riordan. *Environmental science for environmental management*. Routledge, 2014. (Cited on page 1.)

Donald S Ornstein. Guessing the next output of a stationary process. *Israel Journal of Mathematics*, 30(3):292–296, 1978. (Cited on pages 24 and 38.)

Efstathios Paparoditis and Dimitris N Politis. Tapered block bootstrap. *Biometrika*, 88(4):1105–1119, 2001. (Cited on page 17.)

Efstathios Paparoditis and Dimitris N Politis. Resampling and subsampling for financial time series. In *Handbook of financial time series*, pages 983–999. Springer, 2009. (Cited on page 10.)

Andrew Patton, Dimitris N Politis, and Halbert White. Correction to "automatic block-length selection for the dependent bootstrap" by d. politis and h. white. *Econometric Reviews*, 28(4):372–375, 2009. (Cited on pages 17 and 41.)

Martin Peterson. *An introduction to decision theory*. Cambridge University Press, 2009. (Cited on pages 2 and 9.)

Dimitris N Politis. The impact of bootstrap methods on time series analysis. *Statistical Science*, 18(2):219–230, May 2003. (Cited on page 10.)

Dimitris N Politis and Joseph P Romano. A circular block-resampling procedure for stationary data. *Exploring the limits of bootstrap*, pages 263–270, 1992. (Cited on page 17.)

Dimitris N Politis and Joseph P Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313, 1994. (Cited on page 17.)

Dimitris N Politis and Halbert White. Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 23(1):53–70, 2004. (Cited on pages 17 and 41.)

John W Pratt. Risk aversion in the small and in the large. *Econometrica: Journal of the Econometric Society*, pages 122–136, 1964. (Cited on pages 3 and 57.)

Carl L Pritchard, PMI-RMP PMP, et al. *Risk management: concepts and guidance*. CRC Press, 2014. (Cited on page 1.)

Ross A Ritchie and Dr Jannis Angelis. Operational risk: From finance to operations management, 2011. (Cited on page 1.)

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952. (Cited on page 50.)

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951. (Cited on pages 49 and 50.)

Ralph Tyrrell Rockafellar. Coherent approaches to risk in optimization under uncertainty. *Tutorials in operations research*, 3:38–61, 2007. (Cited on page 2.)

Cynthia Rudin and Kiri L Wagstaff. Machine learning for science and society. *Machine Learning*, 95(1):1–9, 2014. (Cited on page 1.)

Esther Ruiz and Lorenzo Pascual. Bootstrapping financial time series. *Journal of Economic Surveys*, 16(3):271–300, 2002. (Cited on page 10.)

Boris Ryabko. Prediction of random sequences and universal coding. *Problems of Information Transmission*, 24:87–96, 1988. (Cited on pages 12, 24, 25, 37 and 39.)

Boris Ryabko. Applications of Kolmogorov complexity and universal codes to nonparametric estimation of characteristics of time series. *Fundamenta Informaticae*, 83(06):1–20, 2008. (Cited on pages 12, 24 and 37.)

Boris Ryabko. Compression-based methods for nonparametric prediction and estimation of some characteristics of time series. *IEEE Transactions on Information Theory*, 55:4309–4315, 2009. (Cited on page 24.)

Boris Ryabko. Applications of universal source coding to statistical analysis of time series. *Selected Topics in Information and Coding Theory, World Scientific Publishing*, pages 289–338, 2010. (Cited on page 23.)

Boris Ryabko and Viktor Aleksandrovich Monarev. Experimental investigation of forecasting methods based on data compression algorithms. *Problems of Information Transmission*, 41(1):65–69, 2005. (Cited on page 24.)

Antoine Salomon and Jean-Yves Audibert. Deviations of stochastic bandit regret. In *Proceedings of the 22nd international conference on Algorithmic learning theory (ALT'11)*, pages 159–173, 2011. (Cited on page 52.)

Alexander Schied. Risk measures and robust optimization problems. *Stochastic Models*, 22(4):753–831, 2006. (Cited on page 2.)

Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1287–1295, 2014. (Cited on page 122.)

Yevgeny Seldin, Peter Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, pages 280–287, 2014. (Cited on page 121.)

Paul C. Shields. *The Ergodic Theory of Discrete Sample Paths (Graduate Studies in Mathematics, V. 13)*. Amer Mathematical Society, July 1996. (Cited on page 39.)

Valgerdur Steinthorsdottir, Gudmar Thorleifsson, Patrick Sulem, Hannes Helgason, Niels Grarup, Asgeir Sigurdsson, Hafdis T Helgadottir, Hrefna Johannsdottir, Olafur T Magnusson, Sigurjon A Gudjonsson, et al. Identification of low-frequency and rare sequence variants associated with elevated or reduced risk of type 2 diabetes. *Nature genetics*, 46(3):294–298, 2014. (Cited on page 10.)

Nassim Nicholas Taleb. The black swan: The impact of the highly improbable (the incerto collection). 2007. (Cited on page 10.)

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933. (Cited on page 49.)

William R Thompson. On the theory of apportionment. *American Journal of Mathematics*, pages 450–456, 1935. (Cited on page 49.)

Long Tran-Thanh and Jia Yuan Yu. Functional bandits. *arXiv preprint arXiv:1405.2432*, 2014. (Cited on page 86.)

Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2012. (Cited on page 48.)

Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012. (Cited on page 103.)

Vladimir Vovk. Aggregating strategies. In *Proceedings of the third annual workshop on Computational learning theory (COLT)*, pages 371–386, 1990. (Cited on page 93.)

Manfred Warmuth and Wouter Koolen. Shifting experts on easy data. pages 1295–1298, 2014. (Cited on page 118.)

Manfred K Warmuth and Dima Kuzmin. Online variance minimization. *Machine learning*, 87(1):1–32, 2012. (Cited on pages xii, 89, 90, 101 and 102.)

Allan Herbert Willett. *The economic theory of risk and insurance*. Number 38. New York: The Columbia University Press, 1901. (Cited on page 2.)

John K Williams. Using random forests to diagnose aviation turbulence. *Machine Learning*, 95(1):51–70, 2014. (Cited on page 1.)

Jia Yuan Yu and Evdokia Nikolova. Sample complexity of risk-averse bandit-arm selection. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2576–2582. AAAI Press, 2013. (Cited on page 85.)

Alexander Zimin, Rasmus Ibsen-Jensen, and Krishnendu Chatterjee. Generalized risk-aversion in stochastic multi-armed bandits. *arXiv preprint arXiv:1405.0833*, 2014. (Cited on page 85.)

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 928–936, 2003. (Cited on pages 102, 119 and 120.)

Janis J Zvingelis. On bootstrap coverage probability with dependent data. *Statistics Textbooks and Monographs*, 169:69–90, 2003. (Cited on pages 17 and 41.)